

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

GABRIEL LANDO

**Uma avaliação de desempenho de
implementações *open source* em software de
núcleos de rede 5G**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em
Engenharia da Computação

Orientador: Prof. Dr. Juliano Wickboldt

Porto Alegre
2022

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitora de Graduação: Prof^a. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Engenharia de Computação: Prof. Walter Fetter Lages

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

“I wanna rock and roll all night

And party every day”

— GENE SIMMONS AND PAUL STANLEY

AGRADECIMENTOS

Aos meus pais, Ernesto Antônio Lando e Izidora Justina Bizotto Lando, e à minha irmã, Anelise Lando, que me apoiaram, acreditaram em mim e me deram suporte desde que saí de casa para vir morar sozinho em outro Estado para cursar Engenharia de Computação em uma das melhores universidades do país. Muito obrigado.

À Isadora, minha parceira de vida, que esteve ao meu lado nesses últimos anos, me apoiando, incentivando e ouvindo eu reclamar de tudo. Muito obrigado, Isa, por toda essa parceria, conselhos e ajuda que você me deu. Eu evoluí muito como pessoa ao seu lado.

Ao Felipe Lando, meu irmão, que me acolhe desde as tentativas falhas do vestibular da UFRGS até hoje, juntamente com a minha cunhada, Amália Machado. Muito obrigado por todos esses anos de parceria, muitos almoços, jantãs e projetos que fizemos juntos desde que eu vim morar na Capital. À Amália, um agradecimento especial por toda a ajuda que você me deu durante o desenvolvimento deste trabalho. Se eu estou aqui hoje é principalmente graças a vocês dois e a Acadêmica Pesquisa.

Ao Urso, meu cachorro, que esteve ao meu lado durante todo o desenvolvimento deste trabalho, seja me mordendo, latindo, pulando por cima de mim querendo brincar ou só deitado em silêncio me fazendo companhia. Eu sei que você não entende a importância do que eu estava fazendo, mas muito obrigado pelo carinho.

Aos meus colegas de trabalho da HP, que compreenderam as diversas vezes que precisei me ausentar para poder realizar atividades referentes a este trabalho. Aqui vai um agradecimento especial à Thayná Minuzzo, que além de colega de trabalho, também foi minha colega durante a graduação, me ajudando em diversas disciplinas, incluindo dicas durante a realização do presente trabalho. Vocês também tem uma parte desta minha conquista. Muito obrigado.

Aos meus colegas e amigos que estão comigo desde a primeira semana de aula, como a Ana e a Narumi, o Fischer, o Probst e o Rodrigo, além de outras amizades que eu fiz durante todos esses anos de graduação. Vocês me confortaram, muitas vezes indiretamente, só de saber que todos nós estávamos passando pelas mesmas dificuldades durante toda a graduação, incluindo nesse trabalho.

Aos meus padrinhos e primos, Tia Enedina, Tio Ronei, Leonardo, Ramon, Lucas e Marina, que sempre me incentivaram e me acolheram nos feriados e finais de semana. Muito obrigado por esse tempo juntos, me ajudaram a esfriar a cabeça nos momentos

mais difíceis da graduação.

Ao meu orientador, Prof. Dr. Juliano Wickboldt, que me ofereceu a primeira oportunidade de bolsa de pesquisa, lá em 2017, no meu segundo semestre de graduação, com o Projeto FUTEBOL e, em 2021, aceitou ser meu orientador para o desenvolvimento desta pesquisa. Aqui também vão meus agradecimentos aos bolsistas de iniciação científica do projeto PORVIR-5G, Lucas Schierholt e Mateus Milesi, que me ajudaram no desenvolvimento deste trabalho. Sem vocês, eu não teria chegado até aqui.

Por fim, agradeço a todos aqueles que, de alguma forma, contribuíram para que este trabalho fosse concluído. Espero que os resultados obtidos ajudem a desenvolver essa área da computação.

RESUMO

Com o advento das redes 5G, surgem implementações de código aberto de seus núcleos, que devem seguir uma especificação descrita pela 3GPP. Para validar a implementação, é necessário realizar testes sobre esses núcleos de rede 5G. O presente trabalho desenvolve uma prova de conceito de um módulo para execução de testes de desempenho em implementações em código aberto de núcleos de redes 5G. O objetivo deste trabalho é analisar como se comportam as diferentes implementações de código aberto dos núcleos de rede 5G, *free5GC*, *Open5GS* e *OpenAirInterface*, para a execução de procedimentos em escala. Para responder a esse problema, primeiramente foi feito um breve resumo sobre a evolução das redes móveis, com foco em implementações em software de núcleos de redes 5G. Após, uma arquitetura de software foi desenvolvida, criando um módulo de extensão para o testador *my5G-RANTester* para realizar testes de desempenho nessas implementações de código aberto de núcleos de redes 5G. Foram executados dois distintos experimentos sobre essas implementações, sendo eles medindo o tempo de registro de cada equipamento de usuário e medindo a largura de banda do plano de dados entre o equipamento de usuário e o núcleo da rede. Dentre os principais resultados obtidos, foi possível observar que o *free5GC* apresenta melhor desempenho em relação à largura de banda disponível para os equipamentos de usuário, enquanto que o *Open5GS* apresenta mais estabilidade durante o processo de registro de múltiplos equipamentos de usuário. Este trabalho possui duas importantes contribuições teóricas para a literatura. A primeira contribuição é a agregação de conhecimento sobre testes de desempenho em redes móveis, enquanto a segunda é relacionada com a estabilidade e limitações das implementações de núcleos 5G em software. A principal contribuição prática deste trabalho é o desenvolvimento de um módulo para realizar testes de desempenho em núcleos de rede 5G. **Palavras-chave:** Redes móveis. Núcleo 5G. Teste de desempenho. *Open5GS*. *free5GC*. *my5G-RANTester*.

A method for evaluating the performance of open source software implementations of 5G network cores

ABSTRACT

With the advent of 5G networks, open source implementations of their cores appear, which must follow a specification described by 3GPP. To validate the implementation, it is necessary to perform tests on these 5G network cores. The present work develops a proof of concept of a module to run performance tests in open source implementations of 5G network cores. The objective of this work is to analyze how the different open source implementations of the 5G network cores free5GC, Open5GS and OpenAirInterface behave, for the execution of procedures at scale. To answer this problem, a brief summary of the evolution of mobile networks was first made, focusing on software implementations of 5G network cores. Afterwards, a software architecture was developed, creating an extension module for the my5G-RANTester tester to run performance tests on these open source implementations of 5G network cores. Two different experiments were performed on these implementations, being them measuring the registration time of each user equipment and measuring the data plane bandwidth between the user equipment and the network core. Among the main results obtained, it was possible to observe that free5GC presents better performance in relation to the bandwidth available to user devices, while Open5GS presents more stability during the registration process of multiple user devices. This work has two important theoretical contributions to the literature. The first contribution is the aggregation of knowledge about performance testing in mobile networks, while the second is related to the stability and limitations of 5G core implementations in software. The main practical contribution of this work is the development of a module to perform performance tests on 5G network cores.

Keywords: Mobile networks. 5G core. Performance tests. Open5GS. free5GC. my5G-RANTester.

LISTA DE ABREVIATURAS E SIGLAS

1G	Primeira geração de redes móveis
2G	Segunda geração de redes móveis
3G	Terceira geração de redes móveis
3GPP	<i>3rd Generation Partnership Project</i>
4G	Quarta geração de redes móveis
5G	Quinta geração de redes móveis
5GC	<i>5G Core</i>
5G NR	<i>5G New Radio</i>
AF	<i>Application Function</i>
AMF	<i>Access and Mobility Management Function</i>
API	<i>Application Programming Interface</i>
AUSF	<i>Authentication Server Function</i>
Bps	Bytes por segundo
CN	<i>Core Network</i>
CPU	<i>Central Processing Unit</i>
eNodeB	<i>Evolved Node B</i>
EPC	<i>Evolved Packet Core</i>
E-UTRAN	<i>Evolved UTRAN</i>
Gbps	Giga bits por segundo
gNB	<i>gNodeB</i>
GSM	<i>Global System for Mobile Communications</i>
HSDPA	<i>High-Speed Downlink Packet Access</i>
HSS	<i>Home Subscriber Server</i>
IoT	<i>Internet of Things</i>

IP	<i>Internet Protocol</i>
kbps	Quilo <i>bits</i> por segundo
KVM	<i>Kernel-based Virtual Machine</i>
LTE	<i>Long Term Evolution</i>
LTE-A	<i>LTE-Advanced</i>
LTE-R	<i>LTE-Railway</i>
Mbps	Mega <i>bits</i> por segundo
MME	<i>Mobility Management Entity</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
ms	Milissegundos
N3IWF	<i>Non-3GPP InterWorking Function</i>
NAS	<i>Non-Access Stratum</i>
NB	<i>Node B</i>
NEF	<i>Network Exposure Function</i>
NGAP	<i>NG Application Protocol</i>
NRF	<i>Network Repository Function</i>
ns	Nanossegundos
NSA	<i>Non-Stand Alone</i>
NSSF	<i>Network Slice Selection Function</i>
NWDAF	<i>Network Data Analytics Function</i>
OAI	<i>OpenAirInterface</i>
PCF	<i>Policy Control Function</i>
PDN GW	<i>Packet Data Network Gateway</i>
PDU	<i>Packet Data Unit</i>
PORVIR-5G	<i>Programmability, Orchestration and Virtualization on 5G networks</i>
PSTN	<i>Public Switched Telephone Network</i>

QoS	Qualidade de serviço
RAN	<i>Radio Access Network</i>
s	Segundos
SA	<i>Stand Alone</i>
SBA	<i>Service-based Architecture</i>
Serving GW	<i>Serving Gateway</i>
SMF	<i>Session Management Function</i>
SMS	<i>Short Message Service</i>
TCP	<i>Transmission Control Protocol</i>
UDM	<i>Unified Data Management</i>
UDP	<i>User Datagram Protocol</i>
UE	<i>User Equipment</i>
UMTS	<i>Universal Mobile Telecommunications System</i>
UPF	<i>User Plane Function</i>
UTRAN	<i>UMTS Terrestrial Radio Network</i>
VoIP	<i>Voice over IP</i>
VoLTE	<i>Voice over LTE</i>
WCDMA	<i>Wide-Band Code-Division Multiple Access</i>

LISTA DE FIGURAS

Figura 2.1 Core 5G software components.....	20
Figura 2.2 5G Core (5GC) Service-based Architecture (SBA).....	21
Figura 4.1 Arquitetura do testador	33
Figura 4.2 Fluxograma de execução do orquestrador	37
Figura 5.1 Tempo de registro e estabelecimento de sessão PDU para cada núcleo 5G ..	42
Figura 5.2 Média e desvio padrão dos erros de conectividade	43
Figura 5.3 Largura de banda para cada configuração de quantidade de UE para o núcleo <i>free5GC</i>	45
Figura 5.4 Largura de banda para cada configuração de quantidade de UE para o núcleo <i>Open5GS</i>	46
Figura 5.5 Valor agregado da largura de banda para cada configuração de quantidade de UEs	47
Figura 5.6 Valor agregado da largura de banda para cada configuração de quantidade de UEs	49
Figura 5.7 Valor agregado da largura de banda para cada configuração de quantidade de UEs	50
Figura 5.8 Valor agregado da largura de banda para cada configuração de quantidade de UEs	51
Figura 5.9 Uso de processador e memória RAM para a execução com atraso de 500 ms entre conexões	53
Figura 5.10 Uso de processador e memória RAM para o experimento de desempenho de plano de dados com 10 UEs conectados	55
Figura 5.11 Uso de processador e memória RAM para o experimento de desempenho de plano de dados com 4 UEs conectados	57

LISTA DE TABELAS

Tabela 3.1 Tabela comparando os trabalhos relacionados	26
Tabela 5.1 Configuração da máquina de testes	38
Tabela 5.2 Tabela detalhando os resultados dos experimentos	44

SUMÁRIO

1 INTRODUÇÃO	14
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 Primeiras gerações de redes móveis	17
2.2 Redes 5G	19
2.2.1 Protocolos NAS e NGAP e Sessões de PDU	21
2.2.2 Principais componentes da Rede 5G.....	22
2.3 Testes em rede móveis definidas por software	23
2.3.1 Testes de conformidade.....	23
2.3.2 Testes de robustez	24
2.3.3 Testes de desempenho.....	24
3 TRABALHOS RELACIONADOS	26
3.1 Tutorial on communication between access networks and the 5G core	26
3.2 Realizing 5G Network Slicing Provisioning with Open Source Software	28
3.3 Delay metrics and delay characteristics: A study of four Swedish HSDPA+ and LTE networks	29
3.4 IoTBench: Towards a Benchmark for Low-power Wireless Networking	30
3.5 Performance Test of LTE-R Railway Wireless Communication at High-Speed (350 km/h) Environments	31
4 SOLUÇÃO	32
4.1 Contextualização	32
4.2 Arquitetura	32
4.2.1 Módulo de teste.....	34
4.2.2 Módulo de coleta de dados do testador.....	35
4.2.3 Módulo de coleta de dados do núcleo.....	35
4.2.4 Orquestrador	36
5 ANÁLISE DOS DADOS COLETADOS	38
5.1 Infraestrutura e parâmetros	38
5.1.1 Testes de tempos de registro e estabelecimento de sessão do núcleo 5G	39
5.1.2 Testes de desempenho do plano de dados do núcleo 5G	40
5.2 Análise e discussão dos tempos de registro e estabelecimento de sessão do núcleo 5G	41
5.3 Análise e discussão dos resultados em relação ao desempenho do plano de dados	44
5.4 Análise e discussão em relação ao consumo de recursos	51
5.4.1 Análise e discussão em relação ao consumo de recursos do experimento de registro e estabelecimento de sessão do núcleo 5G	52
5.4.2 Análise e discussão em relação ao consumo de recursos do experimento de desempenho do plano de dados.....	54
6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	59
REFERÊNCIAS	61

1 INTRODUÇÃO

A quinta geração de redes móveis (5G) teve sua especificação inicial disponibilizada em 2017 na *Release 15* da *3rd Generation Partnership Project* (3GPP) (REDANA et al., 2020), trazendo diversas melhorias em relação à quarta geração de redes móveis (4G). É possível citar como principais melhorias da 5G em relação às gerações anteriores as maiores taxas de transferência de dados, a maior área de cobertura, a menor latência de comunicação, o suporte à maior densidade de dispositivos e a redução no consumo de energia (AHMAD et al., 2019).

Para que seja possível atingir as metas propostas, uma reformulação no núcleo da rede se torna necessária por ser considerado o elemento mais crítico da rede 5G (CARDOSO et al., 2020). O núcleo de uma rede 5G é responsável por estabelecer uma conexão estável e segura entre dispositivos e prover acesso aos serviços disponibilizados pela rede móvel.

Com o objetivo de reduzir os custos de implantação e manutenção das redes móveis, iniciou-se o desenvolvimento de implementações em software do seu núcleo. Implementações em software desse componente permitem que atualizações sejam feitas de forma simplificada, possibilitando a correção de falhas descobertas durante a operação da rede. Além disso, facilita a evolução da rede para futuras gerações, sem a necessidade de realizar deslocamento para os locais de instalação dos equipamentos, visto que a atualização pode ser feita de forma remota. O hardware utilizado para a implantação dessas redes em software é um equipamento de propósito geral, fabricado em larga escala e com custos reduzidos (CARDOSO et al., 2020).

A implementação de redes móveis privadas, sem vínculo com operadoras, se tornou uma opção para simplificar a implantação de redes sem fios em larga escala em ambientes corporativos, principalmente pela facilidade de implementação em máquinas convencionais. O conceito de redes móveis privadas surgiu com o advento da Indústria 4.0, que visa transformar a manufatura industrial por meio da digitalização e exploração das potencialidades das novas tecnologias (ROJKO, 2017). Uma indústria pode criar e gerenciar sua própria rede 4G ou 5G, permitindo a conexão de milhares de dispositivos e sensores, provendo segurança, performance e baixa latência. Existem implementações *open source* em software de estações de rádio e núcleos de redes 4G e 5G, como *free5GC*¹,

¹<<https://www.free5gc.org>>

*Open5GS*², *OpenAirInterface*³ (OAI), *srsRAN*⁴, dentre outras, facilitando a implantação desse tipo de rede móvel em ambientes corporativos.

Uma vez que deseja-se implementar uma rede móvel privada, é preciso avaliar o cenário para identificar a melhor relação custo-benefício. Sendo assim, é preciso verificar, principalmente, a área de cobertura a ser atendida pelas antenas, quantos dispositivos serão conectados, qual o tráfego de dados que será gerado e qual a latência máxima desejada para a comunicação dos dispositivos. Após coletar essas informações, pode-se definir o tipo de equipamento que precisa ser adquirido para suprir a necessidade dessa rede. Entretanto, para que se possa definir qual o hardware a ser usado, é preciso saber como as implementações *open source* se comportam ao serem executadas nesses equipamentos.

Nesse contexto, estão inseridos os testes para averiguar o comportamento dessas redes móveis em diversos cenários. Como as implementações *open source* não possuem garantias, é preciso executar testes de conformidade e robustez, por exemplo, para garantir que o núcleo da rede se comporte da forma que foi especificado. Além disso, é preciso aferir o desempenho das diversas implementações, permitindo avaliar qual hardware seria capaz de rodar a implementação desejada de acordo com a capacidade da rede que se deseja implementar.

Desta forma, o presente trabalho tem como objetivo responder a seguinte questão: como se comportam as diferentes implementações *open source* de núcleo 5G para execução dos procedimentos de registro e qual a largura de banda disponível para uso em escala? Esse trabalho aplica o teste de desempenho em algumas das principais implementações de código aberto de núcleos de rede 5G. Para isso, é proposto e implementado um módulo de extensão do testador *my5G-RANTester*⁵ para executar testes de desempenho sobre essas implementações de núcleo 5G.

Quanto às contribuições teóricas, este trabalho agrega conhecimento em relação a testes de desempenho, sobretudo em núcleos de redes 5G, trazendo discussões relevantes para a área de desenvolvimento de software. Também apresenta as limitações encontradas nas implementações testadas, mostrando a maturidade dessas implementações para uso em escala. Em relação às contribuições práticas, este trabalho desenvolve um módulo para ser acoplado em um testador de código aberto de núcleos de rede 5G.

O restante do presente trabalho foi dividido nos seguintes capítulos. O capítulo

²<<https://open5gs.org>>

³<<https://openairinterface.org>>

⁴<<https://www.srslte.com>>

⁵<https://github.com/my5G/my5G-RANTester>

de fundamentação teórica descreve a evolução das redes móveis e traz definições sobre testes em software. No capítulo de trabalhos relacionados, são descritos os estudos que se assemelham com o presente trabalho, além de demonstrar como este trabalho difere da literatura atual. O capítulo de solução demonstra a arquitetura desenvolvida para execução de experimentos de testes de desempenho, coleta e análise de dados. No capítulo de análise dos dados coletados, são discutidos os resultados dos experimentos, demonstrando o desempenho e as limitações encontradas nos núcleos testados. Por fim, nas considerações finais, são apresentadas com mais detalhes as contribuições teóricas e práticas, as limitações encontradas e sugestões de futuros trabalhos a partir destas descobertas.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma visão geral sobre as redes móveis, desde a primeira até a 5G, abordando os componentes da rede e detalhando as segmentações do núcleo presente nas últimas gerações. Também são descritos os dois principais protocolos de comunicação utilizados nas redes 5G. Por fim, são apresentados diferentes tipos de testes e sua importância sobre as redes móveis definidas por software.

2.1 Primeiras gerações de redes móveis

Conhecer a evolução das redes móveis é necessário devido à complexidade das redes modernas. Originalmente, as redes móveis tinham o foco em comunicações de voz analógica, sendo uma extensão da rede de telefonia pública (*Public Switched Telephone Network - PSTN*), que operava com rede de roteamento de circuitos (CARDOSO et al., 2020). Essas redes móveis foram adaptadas para transporte de dados com taxas de transmissão muito baixas, atingindo velocidades de até 2,4 quilo *bits* por segundo (kbps) e ficaram conhecidas como primeira geração de redes móveis (1G) (VORA, 2015).

Com a necessidade de evolução das redes 1G, surge a segunda geração de redes móveis (2G), que provia um sinal com melhor qualidade, maiores taxas de transferência de dados e maior aproveitamento do espectro das ondas de rádio. Essa tecnologia ainda mantinha seu foco em roteamento de circuitos. Nessa geração, também surge o serviço de troca de mensagens curtas, conhecido como *Short Message Service* (SMS). Na 2G, a tecnologia de transporte de dados *Global System for Mobile Communications* (GSM) foi inserida (BHALLA; BHALLA et al., 2010).

Em 1998, inicia-se a parceria entre diversas organizações responsáveis pela criação dos padrões de telecomunicação denominada 3GPP¹, que publica a *Release 1999* no início do ano 2000. Em redes móveis, *releases* são conjuntos de recursos e especificações disponibilizadas a cada período, definindo os padrões a serem seguidos para a implementação dessas redes. A *Release 1999* descreve uma nova tecnologia de redes móveis, com foco em roteamento de pacotes. Essa tecnologia mantinha a compatibilidade com as tecnologias antigas, sendo denominada terceira geração de redes móveis (3G) (3GPP, 2009).

Inicialmente, o padrão definido para as redes 3G foi o *Wide-Band Code-Divison*

¹<https://www.3gpp.org/about-3gpp>

Multiple Access (WCDMA). Em 2002, o início do desenvolvimento do padrão *High-Speed Downlink Packet Access* (HSDPA) é anunciado, suportando velocidades de transmissão teóricas de até 14 mega *bits* por segundo (Mbps). Isso permitiria a navegação na internet mais rápida, acesso a conteúdos de televisão diretamente no aparelho celular e chamadas de vídeo. O padrão HSDPA foi chamado de 3.5G e teve a sua especificação finalizada em 2004 (LAMBA; YADAV; DEVI, 2012).

No final de 2008, a *Release 8* é publicada pela 3GPP, especificando os requisitos para a tecnologia *Long Term Evolution* (LTE), que substituiria a terceira geração de redes móveis. Entretanto, essa especificação não preenchia os requisitos mínimos para ser considerada uma 4G, sendo conhecida como 3.9G (PERAL-ROSADO et al., 2018). Dentre as mudanças provindas desta nova tecnologia, é importante mencionar a remoção do suporte à roteamento de circuitos, fazendo com que essa geração usasse apenas roteamento de pacotes sobre *Internet Protocol* (IP) para todo o tráfego de informações na rede.

Toda a informação de áudio de ligações telefônicas era trafegada sobre a rede de roteamento de circuitos nas gerações anteriores ao LTE. Com a remoção dessa tecnologia no LTE, foi necessário desenvolver um novo protocolo, surgindo assim o *Voice over LTE* (VoLTE). O VoLTE, assim como a tecnologia *Voice over IP* (VoIP), trafega os dados de voz de ligações telefônicas sobre a rede de roteamento de pacotes, permitindo que as operadoras de redes móveis pudessem continuar oferecendo o serviço de ligações telefônicas para seus clientes. Nesta época, esse serviço era o mais rentável para as operadoras (YI et al., 2012).

Visando preencher os requisitos mínimos para a criação da 4G, no início de 2011 é publicada a *Release 10* pela 3GPP, criando assim a tecnologia *LTE-Advanced* (LTE-A) (3GPP, 2022e). A 4G prometia velocidades de *download* de pico de até 1 giga *bits* por segundo (Gbps) e velocidades de *upload* de pico de até 100 Mbps. Atingir tais velocidades só era possível graças à remoção do suporte ao roteamento de circuitos que ocorreu no 3.9G.

Antes do surgimento do LTE, existiam apenas 3 componentes na rede. Um era o equipamento de usuário, denominado *User Equipment* (UE), sendo esse o dispositivo móvel. Outro componente era a rede de acesso via rádio, ou *Radio Access Network* (RAN), denominada na tecnologia 3G de *Universal Mobile Telecommunications System* (UMTS) *Terrestrial Radio Network* (UTRAN), que englobava a estação de rádio base, chamada de *Node B* (NB). Por fim, o terceiro componente era o núcleo da rede, ou *Core Network* (CN), que na 3G era denominado *UMTS Core Network* (MIAH; TAN, 2002). No entanto, a es-

pecificação da rede LTE trouxe uma melhor separação dos seus componentes do núcleo da rede, denominado *Evolved Packet Core* (EPC). A RAN foi renomeada para *Evolved UTRAN* (E-UTRAN), chamando a estação base de *Evolved Node B* (eNodeB). O EPC foi separado em quatro elementos, sendo eles o *Serving Gateway* (*Serving GW*), o *Packet Data Network Gateway* (*PDN GW*), o *Mobility Management Entity* (MME) e o *Home Subscriber Server* (HSS) (3GPP, 2021a). Esses quatro elementos foram posteriormente subdivididos e serviram de base para a criação da rede 5G.

2.2 Redes 5G

Em 2017, a 3GPP publica a versão inicial da *Release 15*, com as definições da 5G (3GPP, 2020). Assim como nas gerações anteriores, a 5G visa introduzir melhorias na capacidade de transmissão de dados da RAN e redução da latência de comunicação entre o UE e a RAN. Essa geração tem como objetivo criar uma rede mais dinâmica e com maior flexibilidade em relação ao LTE. Nessa geração inicia-se a integração com tecnologias de comunicação sem fios não-3GPP, que enquadram dispositivos denominados de *Internet of Things* (IoT).

A *Release 15* também introduz as arquiteturas de rede 5G *Non-Stand Alone* (NSA) e 5G *Stand Alone* (SA). Na arquitetura NSA, o núcleo da rede 4G é utilizado para a autenticação e comunicação dos dispositivos 5G, substituindo a E-UTRAN pela *5G New Radio* (5G NR) e, conseqüentemente, a eNodeB pela gNodeB (gNB). A rede NSA é considerada uma rede de transição entre a 4G e a 5G, pois o custo de implementação é baixo em ambientes onde já existe cobertura 4G das operadoras.

Por outro lado, a arquitetura SA introduz um novo núcleo da rede, denominado de *5G Core* (5GC). De acordo com a especificação da 3GPP, o 5GC é um conjunto de componentes interconectados através de uma camada de serviços. Cada componente tem seu grupo específico de responsabilidades por consumir e prover serviços de e para outros elementos do sistema 5G, através de uma *Application Programming Interface* (API) (3GPP, 2020).

A rede 5G também trouxe uma nova arquitetura do núcleo da rede, ilustrada em alto nível na Figura 2.1. A rede é formada por 12 componentes. Os três principais componentes dessa rede são o *Access and Mobility Management Function* (AMF), o *Session Management Function* (SMF) e o *User Plane Function* (UPF). O AMF é responsável por garantir que o processo de comunicação ocorra de maneira coesa e transparente, conside-

rando a mobilidade do usuário como principal fator. O SMF é responsável por estabelecer, modificar e liberar as sessões de cada UE, além de requisitar a alocação de um endereço de IP para esses UEs. O UPF é o responsável pelo processamento e encaminhamento dos dados provindos dos UEs. Esses componentes serão discutidos com maior profundidade na subseção 2.2.2.

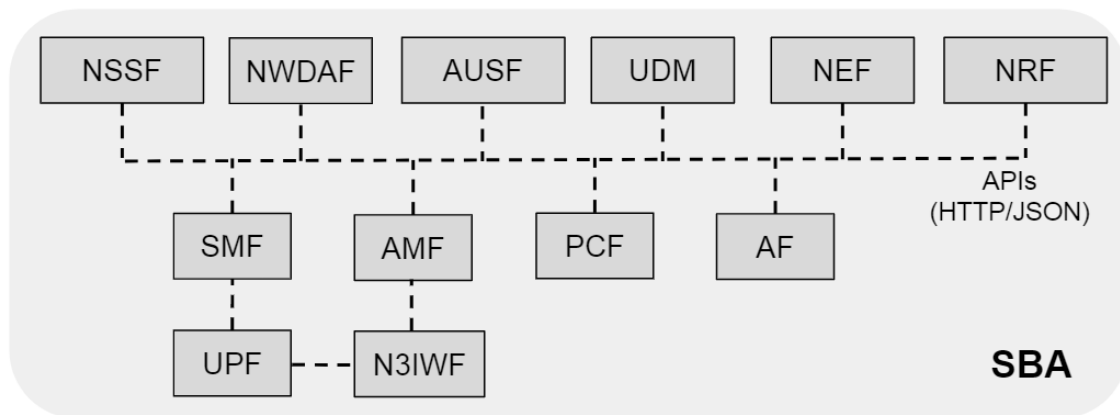


Figura 2.1 – Core 5G software components.

Fonte: (CARDOSO et al., 2020, p. 15)

Os demais nove componentes do núcleo da 5G realizam uma variedade de funções. O *Authentication Server Function* (AUSF) é a função responsável por prover os serviços de autenticação dos UEs. O *Unified Data Management* (UDM) gerencia as informações dos usuários conectados na rede. O *Network Repository Function* (NRF) é um repositório que lista todas as funções de rede disponíveis em sua instância do núcleo do 5G. O *Policy Control Function* (PCF) é o responsável por controlar o comportamento da rede, aplicando políticas de segurança e controle. O *Network Slice Selection Function* (NSSF) é o componente responsável por controlar os *slices* da rede para cada UE. O *Network Exposure Function* (NEF) é responsável por expor eventos internos relacionados aos UEs. O *Network Data Analytics Function* (NWDAF) é a função responsável por coletar e analisar dados provindos dos outros componentes da rede, incluindo informações dos usuários. O *Application Function* (AF) é um componente genérico que interage com os outros componentes no intuito de melhorar a qualidade do serviço para o usuário. O *Non-3GPP InterWorking Function* (N3IWF) é o responsável por integrar dispositivos não-3GPP com a rede 5G.

2.2.1 Protocolos NAS e NGAP e Sessões de PDU

Os protocolos *Non-Access Stratum* (NAS) e *NG Application Protocol* (NGAP) são essenciais para a comunicação entre o núcleo da rede, a RAN e o UE através do plano de controle. As sessões de *Packet Data Unit* (PDU) são importantes para garantir o funcionamento do plano de usuário da rede 5G. A Figura 2.2 representa a arquitetura de uma rede 5G baseada em serviços (*Service-based Architecture* - SBA), exibindo os protocolos NAS e NGAP e as sessões de PDU entre o UE, a gNB e o núcleo da rede.

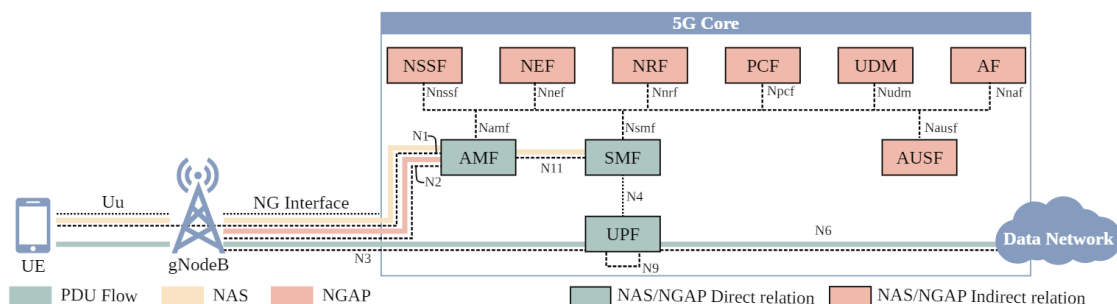


Figura 2.2 – 5G Core (5GC) Service-based Architecture (SBA).

Fonte: (SILVEIRA et al., 2022, p. 3)

O protocolo NAS é usado para fazer a comunicação entre o UE e a função do núcleo AMF, tanto para dispositivos 3GPP quanto não-3GPP. A principal função do protocolo NAS é dar suporte à mobilidade do UE, incluindo procedimentos como autenticação, identificação e atualização das configurações genéricas do UE. O NAS também é responsável por suportar os procedimentos de gerenciamento de sessões, para estabelecer e manter a conexão de dados entre o UE e a rede de dados (3GPP, 2022c). Esse protocolo também é o responsável por fazer a autenticação e gerenciar a conexão da gNB com o núcleo da rede.

O protocolo NGAP é o protocolo padrão para a comunicação do plano de controle entre a RAN e o núcleo da rede. O NGAP suporta os procedimentos de gerenciamento de interfaces, transporte das mensagens do protocolo NAS, gerenciamento de contexto do UE e das sessões de PDU. O gerenciamento de interfaces é o responsável por estabelecer e manter a conexão entre os componentes do núcleo da rede, onde as mensagens dos protocolos NAS e NGAP são transportadas. O Protocolo NGAP é responsável por encapsular e transportar as mensagens NAS, provindas do UE, entre a RAN e o AMF. O gerenciamento de contexto do UE provê informações dos UEs para a RAN, como informações de segurança, lista de restrições de mobilidade e capacidade do rádio do UE. O gerenciamento das sessões de PDU é responsável por administrar os fluxos das unidades

de pacotes de dados entre o núcleo da rede e a RAN, estabelecendo a interface de rádio utilizada para tráfego de controle e dados dos UEs (3GPP, 2021b).

As sessões de PDU são responsáveis por prover a conexão fim-a-fim para o plano de usuário entre o UE e a rede de dados externa, através da função de rede UPF. O fluxo de dados trafegado através das sessões de PDU é chamado de fluxo PDU e seus pacotes de dados são encapsulados sobre o protocolo de rede *User Datagram Protocol* (UDP) (3GPP, 2021c).

2.2.2 Principais componentes da Rede 5G

Dentro do núcleo da rede, os componentes AMF, SMF e UPF são os responsáveis por gerenciar todo o tráfego de informações provindos da RAN. O AMF é a porta de entrada do núcleo da rede em relação ao plano de controle. Sendo assim, todo o tráfego dos protocolos NAS e NGAP provindos da RAN é direcionado para essa função do núcleo da rede. Quando uma nova gNB inicia o processo de conexão com um núcleo existente, a autenticação e o estabelecimento da conexão é feito através de mensagens trocadas que utilizam o protocolo NAS entre a RAN e o AMF. Após estabelecida a conexão, uma mensagem é enviada da RAN para o SMF, indiretamente através do AMF, utilizando o protocolo NAS, para que o SMF tenha conhecimento da gNB e possa gerenciar sua sessão.

Estando a RAN operacional, dispositivos 3GPP e não-3GPP dentro da área de cobertura da antena poderão iniciar sua conexão com essa rede 5G. Um dispositivo 3GPP, ao iniciar a conexão com essa rede 5G, troca mensagens com a RAN através do protocolo NAS. A RAN encapsula essas mensagens para o protocolo NGAP e encaminha para o núcleo da rede. O AMF, comunicando-se com os demais componentes da rede, autentica o UE através de um fluxo de informações definido na *Release 15* do 3GPP (3GPP, 2022a). Após concluir a autenticação do UE, o SMF se torna o responsável por gerenciar a sessão do UE. Um fluxo de dados PDU é estabelecido entre o UE e a função do núcleo UPF, que é responsável por gerenciar e encaminhar os pacotes de dados provindos dos UEs.

O UPF é a função do núcleo da rede 5G responsável por ser o ponto de conexão do núcleo da rede com qualquer rede externa. É o UPF que encaminha os pacotes de dados provindos do UE através das sessões de PDU para a rede externa, sendo o responsável pelo roteamento de pacotes entre a rede externa e os diversos UEs conectados nessa rede 5G. Essa função do núcleo da rede 5G aloca os endereços de IP para os UEs quando requisitado pelo SMF (3GPP, 2022d). O UPF também é responsável por coletar métricas

de qualidade de serviço (QoS) do plano de usuário que serão usadas na função NWDAF (3GPP, 2022b). Esses componentes podem ser implementados em software. Para avaliar a implementação, são utilizados testes para verificação e validação da especificação.

2.3 Testes em rede móveis definidas por software

Segundo (BERTOLINO, 2003, p. 2, tradução nossa), “teste de software consiste na verificação dinâmica do comportamento de um programa em um conjunto finito de casos de teste, adequadamente selecionados do domínio de execuções geralmente infinitas, em relação ao comportamento esperado especificado”. Com relação ao comportamento das redes sem fios implementadas em software, é importante executar testes para garantir que o funcionamento da rede esteja dentro do esperado. Dentre os tipos de testes que podem ser executados em cima de implementações de redes 5G *open source*, os que se destacam são os testes de conformidade, robustez e desempenho, como descrito em Silveira et al. (2022) e Zhang et al. (2019).

2.3.1 Testes de conformidade

De acordo com Sarikaya (1989), teste de conformidade é definido como a atividade de teste realizada com o objetivo de verificar as capacidades e o comportamento de uma implementação em teste em relação aos requisitos de conformidade fornecidos no padrão de protocolo. Os resultados dos testes de conformidade são do tipo lógico, sendo verdadeiro no caso onde a implementação em teste seja aprovada e falso no caso contrário. Dessa forma, o uso de testes de conformidade em redes 5G é importante para verificar se a implementação da rede móvel se comporta de acordo com a especificação da 3GPP em casos onde o UE execute corretamente as operações descritas.

Rayner (1987) divide o grupo de testes de conformidade em quatro categorias: testes básicos de interconexão, testes de capacidade, testes de comportamento e testes de resolução de conformidade. Os testes básicos de interconexão são feitos para detectar quaisquer casos graves de não conformidade com a especificação do sistema. Os testes de capacidade verificam se as capacidades observáveis da implementação em teste estão de acordo com os requisitos de conformidade estática da especificação. A conformidade estática é executada sobre o código fonte da aplicação. Os testes de comportamento ve-

rificam a implementação da forma mais abrangente possível para ver se a implementação está de acordo com os requisitos de conformidade dinâmica da especificação. A avaliação da conformidade dinâmica é realizada durante ou após a execução da implementação. Os testes de resolução de conformidade examinam em profundidade requisitos específicos na implementação, verificando se a implementação se comporta exatamente como a especificação destes requisitos.

2.3.2 Testes de robustez

Segundo Radatz, Geraci and Katki (1990, p. 64, tradução nossa), robustez é definida como “o grau em que um sistema ou componente pode funcionar corretamente na presença de entradas inválidas ou condições ambientais estressantes”. Os resultados dos testes de robustez também são do tipo lógico, sendo verdadeiro no caso onde a implementação em teste seja aprovada e falso no caso contrário. Sendo assim, o teste de robustez no núcleo de redes móveis verifica o comportamento da rede em situações atípicas. Silveira et al. (2022) dividem o grupo de testes de robustez em subgrupos de testes, que envolve testes de registro, de autenticação e de segurança, além de testes específicos para cada protocolo e função de rede do núcleo que serão detalhados na Subseção 3.1.

2.3.3 Testes de desempenho

Os testes de desempenho são importantes para validar o funcionamento da rede em uma carga alta de trabalho, o que representa uma situação de sobrecarga de uso da rede, o que pode causar mal funcionamento da rede e, em situações mais críticas, uma interrupção total do serviço. Para a execução de testes de desempenho, costuma-se usar aplicações de referência, chamados de *benchmarks*. Os softwares de *benchmark* são aqueles que utilizam os mesmos parâmetros de entrada para avaliar o desempenho de diferentes sistemas ou serviços (BOANO et al., 2018). Sendo assim, o usuário pode comparar as métricas obtidas na execução com métricas coletadas de outras execuções em diferentes sistemas ou serviços e avaliar o desempenho do teste.

Para a avaliação dos testes de desempenho em redes móveis em software, Lee, Lee and Lee (2021) utilizam dois grupos de métricas. O primeiro grupo engloba métricas de desempenho de rede: taxa de transferência de dados, latência e perda de pacotes. O se-

gundo grupo é composto por métricas de desempenho de hardware: carga do processador, tempo de execução e tempo de processador.

Em relação às métricas de desempenho de rede, a taxa de transferência é definida por Radatz, Geraci and Katki (1990, p. 77, tradução nossa) como “a quantidade de trabalho que pode ser realizada por um sistema ou componente de computador em um determinado período de tempo”. Sendo assim, a taxa de transferência de dados em uma rede pode ser caracterizada como a quantidade de dados movida com sucesso de um lugar para outro em um tempo determinado.

Para Radatz, Geraci and Katki (1990, p. 43, tradução nossa), latência é definida como “o intervalo de tempo entre o instante em que uma unidade de controle de instrução emite uma chamada de dados e o instante em que a transferência de dados é iniciada.” Se tratando de redes de computadores, latência se refere ao tempo que um pacote de dados leva para ser gerado no emissor, transmitido pela rede, e recebido e decodificado no receptor.

Perda de pacotes é definido por Bhadra et al. (2015) como a falha da entrega de pacotes de dados a um destinatário específico. Esse tipo de falha tende a afetar os mais diversos tipos de comunicação e podem produzir dados incorretos no destinatário.

Quanto a métricas de desempenho do hardware, a carga do processador, é definida como o número de processos que estão sendo executados por uma *Central Processing Unit* (CPU) ou estão aguardando para serem executados (SEBASTIAN et al., 2014). Patterson and Hennessy (2014, p. 242, tradução nossa) definem tempo de execução como “o tempo total necessário para o computador concluir uma tarefa, incluindo acessos ao disco, acessos à memória, atividades de entrada e saída de dados, sobrecarga do sistema operacional, tempo de execução da CPU e assim por diante”. Já o tempo de processador é definido como somente o tempo que a CPU gasta computando uma tarefa específica (PATTERSON; HENNESSY, 2014). Essas diversas métricas são utilizadas em estudos para analisar o desempenho de redes móveis.

3 TRABALHOS RELACIONADOS

Nesta seção, são analisados alguns estudos publicados que apresentam testes em redes móveis. A Tabela 3.1 mostra um resumo dos trabalhos relacionados.

Tabela 3.1 – Tabela comparando os trabalhos relacionados

Autor	Tecnologias testadas	Tipos de teste	Códigos testados	Protocolos testados
Silveira et al. (2022)	5G	Conformidade e robustez	<i>free5GC</i> , <i>Open5GS</i> e <i>OAI</i>	NAS e NGAP
Lee, Lee and Lee (2021)	5G	Conformidade e desempenho	<i>free5GC</i>	Fluxo PDU
Garcia, Alfredsson and Brunstrom (2015)	HSDPA+ e LTE	Desempenho	Não se aplica	Fluxo PDU
Boano et al. (2018)	IoT	Desempenho	Não se aplica	Não se aplica
Mahn-Suk et al. (2018)	LTE-R	Desempenho	Não se aplica	Não se aplica

3.1 Tutorial on communication between access networks and the 5G core

O estudo publicado por Silveira et al. (2022) cria uma prova de conceito de uma aplicação para testar núcleos *open source* de rede 5G. Para esse estudo, foram escolhidos os grupos de testes de conformidade e robustez. As implementações suportadas por esse testador são *free5GC*, *Open5GS* e *OAI*.

Essa prova de conceito simula uma RAN e um ou mais UEs. O objetivo do estudo é analisar o comportamento dos diferentes núcleos de rede 5G em dois diferentes cenários. O primeiro cenário executa ações de acordo com o que foi especificado pela 3GPP, através de testes de conformidade. O segundo cenário executa ações utilizando informações inválidas, forçando o núcleo da rede a contornar essa situação de acordo com o especificado, o que representa os testes de robustez.

Em relação aos testes de conformidade, são executados 11 diferentes testes, sendo eles teste de registro, autenticação primária e estabelecimento de chave, identificação, transporte, modo de segurança, atualização de configuração genérica do UE, gerenciamento de sessão, gerenciamento de interface, transporte de mensagens NAS, gerenciamento de contexto do UE e gerenciamento de sessão do PDU. Foi utilizada a *Release*

16 do 3GPP como base para a execução dos testes. O núcleo *Open5GS* foi aprovado em todos os testes de conformidade, enquanto que o *OAI* e o *free5GC* não responderam de acordo com a especificação no teste de atualização de configuração genérica do UE.

Em relação aos testes de robustez, foram executados 7 testes, sendo eles teste de registro, autenticação, segurança, seleção SMF, seleção UPF, validação do fluxo NAS e gerenciamento de interface. No teste de registro, dois casos foram avaliados. No primeiro caso, o UE enviou uma requisição de registro sem o identificador do UE, que é um campo mandatório. No segundo caso, o UE enviou a requisição com as informações criptografadas. No teste de autenticação, são realizadas duas operações incorretas. O primeiro teste foi realizado enviando informações inválidas como resposta no fluxo de autenticação, enquanto que o segundo teste foi forçando uma falha de sincronização entre o núcleo e o UE. Para o teste de segurança, foram testadas duas configurações de segurança inválidas, sendo elas responder a uma solicitação de segurança sem enviar um parâmetro obrigatório na resposta e responder à solicitação sem retransmitir as informações de segurança da requisição. Os teste de seleção SMF e seleção UPF representam um UE fazendo uma requisição para estabelecer um fluxo PDU, porém enviando informações mandatórias inválidas. O teste de validação do fluxo NAS visa verificar se o núcleo da rede se comporta da forma esperada caso a requisição de estabelecimento de sessão PDU seja realizada enviando as informações necessárias em uma ordem incorreta. Por fim, o teste de gerenciamento de interface simula o registro de uma nova RAN na rede 5G, porém enviando informações inválidas para o AMF, o qual deve rejeitar essa requisição.

O estudo considera que todos os núcleos testados obtiveram um desempenho satisfatório para a operação dentro da normalidade. Entretanto, dentre os três núcleos analisados, o *OAI* foi considerado pelos autores do artigo como o núcleo com a implementação menos madura até o momento da publicação. Mais detalhes sobre os resultados dos testes de conformidade podem ser vistos na tabela *Table II* de Silveira et al. (2022, p. 10) e para os testes de desempenho, *Table III* de Silveira et al. (2022, p. 11).

Os testes realizados no artigo de Silveira et al. (2022) são de extrema importância para avaliar o funcionamento de um núcleo de rede 5G. Como descrito brevemente no artigo, essa implementação de prova de conceito também suporta a realização de um teste onde múltiplos UEs são conectados no núcleo da rede em forma de fila (realizando uma conexão por vez), afim de avaliar se a implementação em teste do núcleo da rede se comporta de acordo com a especificação em casos onde existam diversos UE conectados na rede. Entretanto, não foram executados testes para comparar o desempenho dessas

implementações de núcleo nos casos em que uma grande quantidade de UEs se conectem ao núcleo de forma paralela, onde múltiplos UEs façam requisições de registro e autenticação em um curto intervalo de tempo. Esse teste é importante para avaliar qual implementação possui a melhor performance em casos de alta carga de trabalho.

3.2 *Realizing 5G Network Slicing Provisioning with Open Source Software*

O artigo de Lee, Lee and Lee (2021) trata sobre fatiamentos de rede 5G utilizando-se da implementação *open source free5GC* do núcleo de rede 5G. A ideia de fatiar uma rede permite criar múltiplas redes lógicas sobre uma única rede física e cada rede lógica tendo suas próprias funções de rede e configurações. Lee, Lee and Lee (2021) executa cinco experimentos, em duas categorias distintas de testes. Primeiramente, executam um experimento de conformidade, verificando se o fatiamento da rede 5G está ocorrendo conforme o especificado. Após, são executados quatro experimentos medindo o desempenho do núcleo dessa implementação de rede 5G utilizando o fatiamento de rede.

Na primeira etapa do trabalho foi executado o primeiro experimento. Este experimento tem como objetivo verificar se os pacotes de dados trafegam pela rede na ordem correta, conforme a especificação. Para isso, utilizaram ferramentas de análise de rede, como o *tcpdump*¹. O teste foi bem sucedido, permitindo que os próximos testes de desempenho pudessem ser realizados.

Com os resultados do primeiro experimento, Lee, Lee and Lee (2021) iniciaram a execução dos quatro experimentos restantes. O segundo experimento teve como objetivo medir a qualidade da rede utilizando aplicações específicas para sobrecarregar a rede de dados e medir a vazão da rede 5G.

O terceiro experimento mediu o desempenho das operações de fatiamento de rede simulando um ambiente de banda larga móvel melhorada, onde o UE executa serviços que geram um alto tráfego de dados na rede. Nesse experimento, foi observado o tráfego de dados em cada dispositivo e fatia da rede.

O quarto experimento simulou uma rede de sensores, onde diversos dispositivos IoT publicaram informações através do protocolo *Message Queuing Telemetry Transport* (MQTT)² e diversas aplicações se inscreveram para receber as mensagens enviadas. Foram coletadas métricas de latência, perda de pacotes, uso de CPU e tempo de execução

¹<https://www.tcpdump.org>

²<https://mqtt.org>

para cada experimento realizado, variando-se a quantidade de dispositivos e aplicações interagindo com a rede. Constatou-se que a latência aumentava gradativamente a cada aumento na quantidade de conexões, havendo em certo momento perda de pacotes devido à sobrecarga da rede.

O quinto experimento consistiu em criar fatias de rede variando-se a quantidade de funções de rede. O objetivo desse experimento foi simular serviços de diferentes complexidades, medindo-se a latência gerada ao adicionar mais funções de rede.

Os experimentos executados por Lee, Lee and Lee (2021) são muito importantes para analisar como a sobrecarga de redes móveis afeta o desempenho de redes 5G fatiadas. Entretanto, esse artigo realizou os experimentos sobre o plano de usuário da rede, não levando em consideração quanto o plano de controle da rede foi afetado.

3.3 Delay metrics and delay characteristics: A study of four Swedish HSDPA+ and LTE networks

O estudo de Garcia, Alfredsson and Brunstrom (2015) realiza testes de desempenho no plano de usuário de redes móveis reais de gerações passadas, sendo elas 3.5G e LTE, com o objetivo de comparar o ganho de desempenho da rede LTE em relação à rede 3.5G em diferentes cenários e operadoras. As métricas coletadas pelo estudo foram latência e largura de banda das redes analisadas. Também foi observado o tempo de carregamento de páginas na internet, simulando o uso real da rede.

Cada teste realizado foi executado em dois diferentes cenários. No primeiro cenário, apenas o serviço usado nos testes estava gerando tráfego de dados na rede móvel, reduzindo-se as interferências externas. No segundo cenário, um tráfego de dados de fundo foi gerado, utilizando o protocolo *Transmission Control Protocol* (TCP) com o intuito de sobrecarregar o adaptador de rede utilizado, simulando um UE real, que possui diversos serviços dividindo os recursos de rede do UE.

Este estudo analisa somente o tráfego de dados em redes móveis, verificando o comportamento das redes em uso normal e com tráfego de fundo. Uma vez que as redes testadas foram redes reais de quatro operadoras locais, entende-se as limitações do estudo e sabe-se que a coleta de métricas relacionadas ao plano de controle da rede não seria possível. Devido a limitações na realização de testes em redes reais, os autores não puderam realizar testes de desempenho durante o processo de ingresso na rede móvel, com intuito de avaliar o plano de controle.

3.4 IoTBench: Towards a Benchmark for Low-power Wireless Networking

O trabalho de Boano et al. (2018) propõe uma arquitetura para o desenvolvimento de uma aplicação de *benchmark* para testes de desempenho em redes sem fios de baixa potência. As redes sem fios de baixa potência são muito importantes para redes de sensoreamento, como as redes de dispositivos IoT. Devido à ausência de aplicações especializadas para a avaliação desse tipo de rede, os autores sugerem uma arquitetura para um testador de desempenho para redes de baixa potência.

O objetivo do trabalho de Boano et al. (2018) é criar uma interface padrão para testes em redes de baixa potência. Os autores consideram como seus principais usuários pesquisadores, empresas e consumidores. Pesquisadores poderiam utilizar essa ferramenta para comparar o desempenho de novos protocolos em relação aos protocolos já disponíveis no mercado. Empresas usariam essa ferramenta para exibir as vantagens de seus protocolos proprietários em relação aos concorrentes. Consumidores poderiam utilizar os dados disponibilizados pela ferramenta para avaliar qual protocolo existente no mercado melhor iria satisfazer as necessidades da rede que se deseja implantar.

Os autores desse trabalho definem três classes de parâmetros e métricas a serem considerados, sendo eles os parâmetros de entrada, as métricas de saída e as métricas observadas. Os parâmetros de entrada são informações providas pelo usuário que são independentes do protocolo a ser testado e irão definir como o testador realizará seus testes. Como exemplo de parâmetros, os autores citam a quantidade de dispositivos simulados na rede ou a carga de tráfego na rede. As métricas de saída são definidas pelos autores como os valores obtidos como resultado dos testes executados. Exemplos de métricas podem ser a taxa de entrega de pacotes ou o consumo de energia do protocolo. Os valores dessas métricas são determinados pelo comportamento do protocolo testado. As métricas observadas são valores obtidos pelo usuário como resultado dos testes executados. Essas métricas caracterizam o ambiente onde os testes foram executados, podendo ser interferências externas, como a quantidade de pessoas presentes durante a execução dos testes.

O trabalho de Boano et al. (2018) não está diretamente relacionado com a proposta do presente trabalho de graduação. Entretanto, a sugestão de arquitetura e métricas utilizadas para a realização de testes de desempenho em redes de baixa potência podem ser reaproveitadas e adaptadas na proposta da presente pesquisa.

3.5 Performance Test of LTE-R Railway Wireless Communication at High-Speed (350 km/h) Environments

Mahn-Suk et al. (2018) descrevem o conceito de *LTE-Railway* (LTE-R) como uma categoria de 4G para operar com UEs em alta velocidade. Nesse estudo, os autores realizam testes sobre uma rede móvel LTE-R em uma linha férrea construída pelo *Korea Railroad Research Institute*, na Coreia do Sul, onde um trem de alta velocidade percorre um trajeto, possuindo trechos ao ar livre e dentro de túnel. Redes LTE-R são importantes para a segurança em viagens de trens, devido à probabilidade de descarrilamentos ou terrorismo, além de melhorar a qualidade dos serviços prestados para os passageiros a bordo.

Neste estudo, são realizados testes de cobertura da rede e potência de sinal, tempo para iniciar uma chamada telefônica, taxa de sucesso de *handover*, taxa de sucesso de recebimento de chamadas, entre outros. Para cada tipo de teste, foram realizadas 5 medições e os resultados foram adicionados em tabelas para melhor visualização. Os testes de cobertura da rede e potência do sinal visam medir a qualidade do sinal no percurso da ferrovia. O teste de tempo para iniciar uma chamada telefônica mede o tempo necessário para se conectar com os serviços de voz para chamadas de emergência. *Handover* é o nome dado ao processo de troca de antenas por um UE, causado principalmente pela mobilidade do UE. Sendo assim, a taxa de sucesso de *handover* mede as interrupções de serviço devido à constante troca de antenas causada pela rápida movimentação do trem. A taxa de sucesso de recebimento de chamadas é medida através da capacidade de um terminal telefônico do trem de receber chamadas durante o percurso do trem. Para cada um dos testes, um critério de aceitação foi definido de acordo com a especificação das redes LTE-R. Todos os testes realizados obtiveram bons resultados de acordo com a especificação. Mais detalhes sobre os testes podem ser vistos em Mahn-Suk et al. (2018).

O trabalho de Mahn-Suk et al. (2018) se relaciona com a proposta do presente trabalho de graduação ao medir o tempo de *handover* nos UEs. O cálculo dessa métrica mede o tempo entre o recebimento do último pacote de dados da antena anterior e o recebimento do primeiro pacote de dados da nova antena (TAYYAB; GELABERT; JANTTI, 2019). Sendo assim, essa métrica inclui o tempo de desconexão de uma antena e conexão da outra, que inclui a latência causada pela autenticação do UE. Essa latência será medida no presente trabalho e o método usado será descrito na Seção 4.

4 SOLUÇÃO

Este capítulo apresenta a arquitetura da solução proposta por essa pesquisa, descrevendo os módulos implementados durante o desenvolvimento deste trabalho.

4.1 Contextualização

Este trabalho faz parte do projeto de pesquisa *Programmability, Orchestration and Virtualization on 5G networks* (PORVIR-5G)¹, um projeto colaborativo entre diversas universidades brasileiras. O principal objetivo desse projeto é conceber uma arquitetura aberta que ofereça programabilidade e orquestração para fatiamento de rede. O projeto de desenvolvimento do testador *my5G-RANTester* tem relação com o projeto de pesquisa PORVIR-5G ao demonstrar a viabilidade da arquitetura desenvolvida em diversos casos de uso explorando requisitos avançados das redes 5G, especialmente em termos de latência, confiabilidade, cobertura, mobilidade e banda. A presente pesquisa teve a contribuição de dois bolsistas de iniciação científica do projeto PORVIR-5G.

4.2 Arquitetura

Uma extensão para a aplicação *my5G-RANTester* foi desenvolvida utilizando-se como base o trabalho de Silveira et al. (2022). Essa prova de conceito teve o intuito de avaliar o comportamento das diferentes implementações de código aberto de núcleo 5G para execução dos procedimentos em escala. O trabalho de Silveira et al. (2022) implementa um testador que tem como objetivo simular e realizar testes sobre os planos de controle e de dados de UEs e de estações de rádio base da rede 5G. O testador foi projetado para realizar testes de conformidade e robustez sobre diferentes implementações de código aberto de núcleos de redes 5G. Todavia, o testador suporta a expansão para outras cargas de trabalhos, permitindo o desenvolvimento de outros tipos de testes, como os testes de desempenho que estão sendo abordados nesta presente pesquisa.

A arquitetura em alto nível da aplicação desenvolvida por Silveira et al. (2022) pode ser vista na Figura 4.1. A prova de conceito, desenvolvida na linguagem de programação *Go*², implementa as camadas de Simulação, do Controlador e da Interface de

¹<https://porvir-5g-project.github.io/>

²Linguagem criada pela Google em 2009 (<https://go.dev>)

usuário. A camada de Simulação imita o comportamento de uma ou mais gNB e de um ou mais UEs. Silveira et al. (2022) simulam em seu estudo múltiplas conexões gNB e UEs em sequência. Na presente pesquisa, a simulação foi realizada com múltiplas conexões gNB e UEs simultâneas com o intuito de validar a performance de cada implementação de núcleo da rede.

A camada do Controlador é a principal camada no funcionamento do testador. Essa camada recebe os comandos do usuário através de uma interface que define qual teste será executado, controla a camada acima e faz a coleta das métricas do experimento. Os planos de testes representam os testes em si que foram escolhidos para serem executados. O Executor recebe as informações do teste, configura as gNB e os UEs e envia as métricas para o Coletor de métricas, que irá armazená-las para futuros usos. O presente trabalho estende o módulo do plano de testes, adicionando a possibilidade de executar testes de conectividade de UEs em paralelo, permitindo medir o comportamento do núcleo da rede operando sobre maior carga de trabalho.

Por fim, a camada da Interface de usuário permite a execução e o monitoramento dos testes, além de exportar as métricas armazenadas na camada acima. É com essa camada que o usuário do testador pode interagir através de parâmetros enviados por uma linha de comandos a ser executada no ambiente de testes.

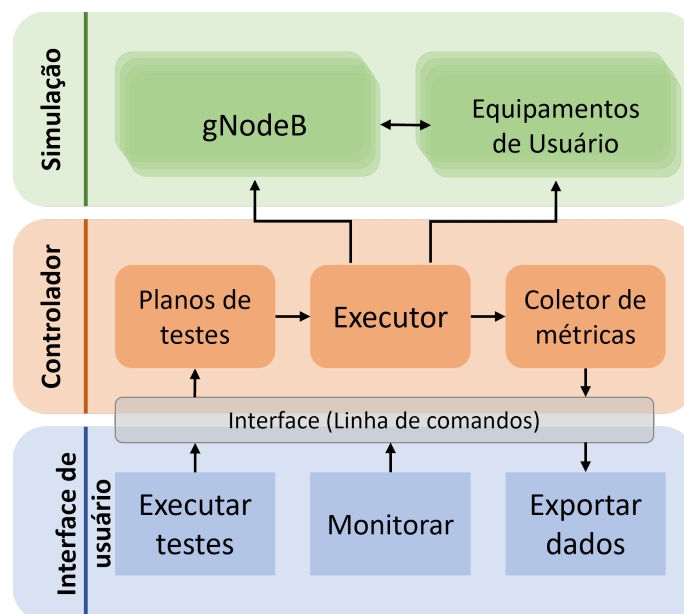


Figura 4.1 – Arquitetura do testador

Para desenvolver este trabalho, além do módulo implementado internamente no testador, também foi necessário desenvolver algumas ferramentas para a automação da criação e configuração do ambiente de testes e da execução dos experimentos. Essas

ferramentas orquestram as múltiplas execuções dos experimentos e coletam os dados gerados para análise futura. A seguir, são apresentadas as ferramentas desenvolvidas no decorrer deste trabalho.

4.2.1 Módulo de teste

Para a execução de testes de desempenho, foi desenvolvido um módulo para o testador que permite a execução de múltiplas conexões de UEs simultâneas. Esse módulo foi implementado suportando parâmetros de configuração para facilitar variações entre os testes. Desta forma, é possível definir o número de UEs a serem conectados, o atraso em milissegundos (ms) entre uma conexão e outra e o atraso em segundos (s) para começar a execução do experimento.

No começo da execução, esse módulo simula uma gNB e inicia a conexão com o núcleo da rede. Após a conexão ser bem sucedida, é aguardado o tempo para iniciar a execução do experimento. Ao iniciar o experimento, o módulo inicia um fluxo de trabalho em paralelo para criar uma nova instância de UE e realizar a conexão com o núcleo da rede. Após iniciar o fluxo de trabalho em paralelo, o módulo aguarda o atraso entre conexões definido pelo usuário antes de iniciar o próximo fluxo, até que a quantidade de dispositivos definida seja atingida.

Durante a etapa de desenvolvimento do módulo, foi descoberta uma limitação na implementação do testador, que não consegue gerenciar mais que 255 UEs conectados simultaneamente. Analisando o código, observa-se que a correção da limitação necessitaria de tempo e recursos não previstos na presente pesquisa. Foi feito um contato com os desenvolvedores do testador sobre essa limitação, mas a solução para o problema não foi implementada até o presente momento. Sendo assim, foi decidido criar múltiplas instâncias do testador, cada uma simulando uma gNB diferente, todas se conectando de forma simultânea na mesma instância do núcleo, cada uma simulando uma quantidade predefinida de UEs. Desse modo, foi possível testar o comportamento de um núcleo de rede 5G com múltiplos UEs e gNB.

4.2.2 Módulo de coleta de dados do testador

O módulo de coleta de dados consiste em um fluxo de execução em paralelo que registra o tempo, em nanossegundos (ns), que cada UE demorou para trocar entre cada estado da conexão com o núcleo da rede 5G. As informações de tempo para cada estado de cada UE são escritas na saída de texto padrão do contêiner utilizado pela instância do testador. Esse módulo foi implementado nesta presente pesquisa, visto que o testador não possuía suporte para a coleta dessa métrica até o presente momento.

Ao final da execução, uma aplicação foi desenvolvida para realizar o processamento das informações geradas através das diversas instâncias do testador, fazendo um pré processamento dos dados, ordenando eles com base no horário da inicialização do UE (em ns a partir do *epoch time*³) e gerando um arquivo de texto separado por vírgulas, permitindo a fácil importação dos resultados através de programas de análise de dados. Inicialmente, essa aplicação foi desenvolvida para coletar e processar os registros de uma única instância do testador, ordenando em ordem crescente pelo identificador do UE os dados no arquivo de saída. Entretanto, foi necessário modificar essa aplicação para suportar múltiplas instâncias de testadores executando em paralelo e reordenar os dados do arquivo de saída, visto que a ordem de conexão dos dispositivos deixou de ser realizada em ordem crescente do identificador.

A aplicação consiste em um *script* escrito na linguagem *JavaScript*⁴ que é executado dentro de um contêiner *Docker*, evitando a instalação de um interpretador de *JavaScript* diretamente na máquina que está executando os experimentos.

4.2.3 Módulo de coleta de dados do núcleo

O desenvolvimento deste módulo teve a contribuição dos bolsistas de iniciação científica do projeto PORVIR-5G. Esse módulo consiste na implementação de um fluxo sobre uma instância da aplicação *Node-RED*⁵. Um fluxo foi desenvolvido para listar todos os contêineres em execução na máquina e realizar a coleta das métricas do *Docker* e armazená-las em uma instância do banco de dados *InfluxDB*⁶. Esse fluxo foi definido

³*Epoch time* é uma unidade de medida de tempo contada a partir da data 1 de janeiro de 1970 às 00:00:00 UTC.

⁴<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

⁵<https://nodered.org/>

⁶<https://www.influxdata.com/>

para ser executado automaticamente após a inicialização do contêiner do *Node-RED*, permitindo a coleta de dados de uso de disco, rede, processador, memória, entre outras, de forma automatizada pelo orquestrador do experimento.

4.2.4 Orquestrador

Com o objetivo de realizar todas as execuções dos experimentos de forma mais semelhante possível, foi desenvolvido um orquestrador que gerencia toda a execução do experimento. O orquestrador consiste em uma série de roteiros de execução, desenvolvidos em sua maioria sobre as linguagens *Bash Script* e *JavaScript*. A Figura 4.2 representa um fluxograma do algoritmo de execução do orquestrador. O orquestrador é executado através de uma interface de usuário por linha de comandos, suportando diversos comandos para definir os parâmetros de execução do experimento, permitindo ao usuário automatizar a execução de experimentos de forma simplificada.

Os parâmetros suportados são a quantidade total de UEs a serem conectados no núcleo da rede, a quantidade de testadores que serão executados em paralelo, o tempo em ms para ser aguardado entre cada conexão de UE, o tempo em s para aguardar a inicialização, qual núcleo da rede 5G deve ser utilizado para executar o experimento e qual experimento será executado. Além dos parâmetros de execução do experimento, o orquestrador também permite o usuário interromper a última execução e limpar o ambiente de testes, removendo todos os contêineres e dados gerados pelo experimento, além de permitir executar o orquestrador em modo de depuração, exibindo para o usuário todos os registros de execução.

Ao iniciar a execução do orquestrador, será feita uma análise sobre o sistema operacional, verificando se a versão necessária do *Kernel* do sistema está instalada. Após realizar esta validação, o orquestrador irá verificar se todas as dependências estão corretamente instaladas e configuradas. Caso contrário, o orquestrador fará as devidas modificações automaticamente. Uma vez que o sistema está pronto, é iniciada a execução do experimento desejado. Primeiramente, o código-fonte do núcleo da rede 5G escolhido será baixado, compilado e seus arquivos binários serão armazenados em uma imagem de contêiner *Docker*, que será iniciada na sequência. Após a inicialização do núcleo, a base de dados desse núcleo é preenchida com as informações necessárias para a conexão dos UEs de acordo com as configurações do experimento. Concluindo-se essa etapa, é iniciado o módulo de coleta de dados do núcleo, descrito anteriormente. Por fim, inicia-se o

contêiner do testador com os parâmetros definidos pelo usuário.

O orquestrador foi desenvolvido para suportar diferentes implementações de núcleos da rede 5G. Para isso, foi desenvolvida uma interface genérica para a inicialização do núcleo. Assim, para adicionar o suporte a um novo núcleo, basta implementar essa interface para o núcleo desejado e adicioná-la ao orquestrador. Para a realização deste trabalho, foi desenvolvido o suporte para as implementações de núcleo 5G *free5GC*, *Open5GS* e *OAI*. Todos os códigos fontes utilizados no desenvolvimento da presente pesquisa encontram-se no repositório do projeto PORVIR-5G⁷ disponível através da plataforma *GitHub*.

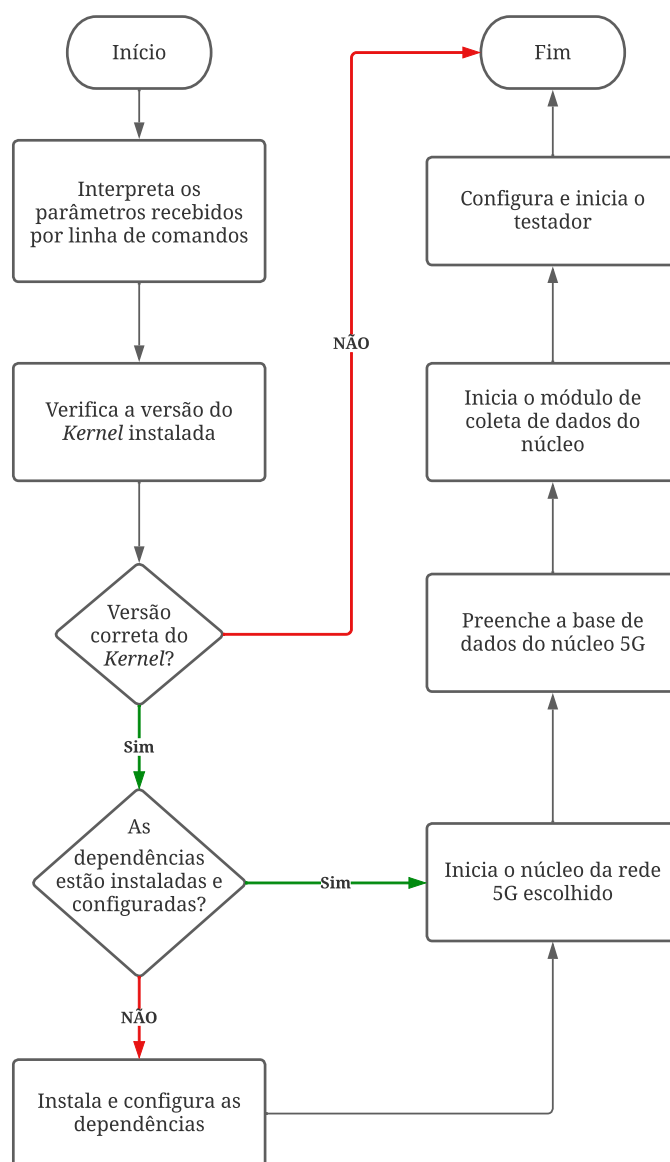


Figura 4.2 – Fluxograma de execução do orquestrador

⁷<https://github.com/PORVIR-5G-Project>

5 ANÁLISE DOS DADOS COLETADOS

Este capítulo introduz a infraestrutura para realização dos testes, quais foram os experimentos realizados e as métricas utilizadas. Também apresenta os resultados dos experimentos e as conclusões geradas a partir da análise dos dados coletados.

5.1 Infraestrutura e parâmetros

Para a realização dos experimentos e validação dos resultados foi utilizada uma máquina virtual hospedada no software *Kernel-based Virtual Machine* (KVM) sobre o sistema operacional *TrueNAS Scale*¹. A máquina virtual executa o sistema operacional Ubuntu Server 20.04.4 LTS e possui diferentes configurações de processador e memória RAM, variando entre 4, 6, 8 e 12 núcleos virtuais de um processador *Intel Core i7 8700T*, 4 e 8 GB de memória RAM DDR4-2666 e um disco virtual de 64 GB armazenado em um SSD M.2 NVMe. Detalhes resumidos da máquina virtual podem ser consultados na Tabela 5.1.

Para a execução dos experimentos, foi criada uma imagem de base da máquina virtual pré-configurada, que era restaurada ao fim de cada execução do experimento, evitando que resquícios de uma execução anterior pudessem afetar outras execuções do experimento. Além disso, também foi utilizado o sistema de containerização *Docker*² para isolar os componentes do núcleo da rede 5G e do testador.

Tabela 5.1 – Configuração da máquina de testes

	Máquinas Virtuais			
CPU	Intel Core i7 8700T @ 2.4 GHz			
	4 núcleos	6 núcleos	8 núcleos	12 núcleos
Memória RAM	DDR4 2666 MHz			
	4 GB	8 GB	8 GB	8 GB
Armazenamento	SSD M.2 64 GB NVMe			
Sistema Operacional	Ubuntu Server 20.04.4 LTS			

Dois experimentos distintos foram realizados sobre diferentes configurações dessa infraestrutura. Os experimentos foram os testes de tempos de registro e estabelecimento de sessão do núcleo 5G e testes de desempenho do plano de dados do núcleo 5G.

¹<https://www.truenas.com/truenas-scale/>

²<https://www.docker.com/>

5.1.1 Testes de tempos de registro e estabelecimento de sessão do núcleo 5G

O experimento para medição de tempos de registro e estabelecimento de sessão do núcleo 5G consiste na simulação de múltiplos UEs tentando se conectar e estabelecer uma sessão PDU com o núcleo 5G a ser testado. Desse modo, é possível medir a latência entre cada estágio da conexão e sua variação de acordo com a carga de trabalho aplicada sobre o núcleo. Para que fosse possível medir essa latência em diferentes cenários de forma automatizada, foi adicionado parâmetros de configuração tanto no orquestrador do experimento quanto no módulo de testes. Neste experimento, também foram coletadas as métricas de uso de processador, memória RAM, disco e rede de todos os componentes do núcleo da rede 5G e do testador, utilizando-se as ferramentas para exportação de métricas dos contêineres *Docker*.

Para esse experimento, foi utilizada a configuração da máquina virtual com 12 núcleos virtuais e 8 GB de memória RAM. Em relação às configurações do experimento, foram utilizadas as versões dos núcleos 5G *free5GC* v3.2.1, sendo a mais recente disponível no momento da execução dos experimentos e *Open5GS* v2.3.6, que é a versão recomendada pelos desenvolvedores do testador. Para cada núcleo, foram usadas as seguintes configurações do testador: número de gNB variando de 1 a 11, com passo de 2, intervalo entre conexões de 100 ms até 500 ms, com passo de 100 ms e cada gNB conectando 100 UEs. Esse experimento foi executado 10 vezes, para obter-se uma quantidade de dados relevantes, onde ruídos externos ao experimento fossem amenizados.

Durante os testes para a realização desse experimento, algumas limitações foram encontradas e devem ser ressaltadas. Primeiramente, pretendia-se utilizar a versão do núcleo *free5GC* v3.0.6, que é recomendada pelos desenvolvedores do testador. Entretanto, essa versão do núcleo apresentou falhas após conectar aproximadamente 10 UEs, o que inviabilizaria a execução desse experimento. O núcleo OAI apresentou uma limitação parecida tanto na versão v1.3.0, que é recomendada para o testador, quanto na versão v1.4.0, que é a mais recente disponível até o presente momento, falhando após conectar exatamente 15 UEs. O núcleo *Open5GS* na versão v2.3.6 consegue conectar exatamente 1024 dispositivos simultâneos antes de apresentar erros na função de rede AMF. Sendo assim, foi decidido que o limite de gNBs seria 11, com o total de 1100 UEs sendo conectados, atingindo a limitação desse núcleo.

5.1.2 Testes de desempenho do plano de dados do núcleo 5G

O experimento para medição do desempenho do plano de dados do núcleo 5G consiste na simulação de 1 ou mais UEs realizando o registro no núcleo da rede e iniciando um plano de dados. Após o registro ser concluído, é medida a capacidade do plano de dados utilizando-se a ferramenta *iPerf*³ versão 2.0.

Para a execução desse experimento, foi decidido utilizar a versão 2.0 da ferramenta *iPerf* ao invés da versão mais recente devido a capacidade da versão 2.0 suportar múltiplas conexões de clientes em uma única instância do servidor, além de permitir exportar as métricas do experimento em um arquivo de texto separado por vírgula, facilitando o pós processamento dos dados. Em relação à máquina virtual utilizada, foram feitos testes variando o número de núcleos virtuais de processador entre 4, 6, 8 e 12 núcleos virtuais e variando a capacidade de memória RAM entre 4 GB e 8 GB. Cada variação na quantidade de núcleos virtuais possuía uma única configuração de memória RAM. Os detalhes das máquinas virtuais estão declarados na Tabela 5.1. Nesse experimento, a principal métrica a ser observada é a largura de banda, em *Bytes* por segundo (Bps), representando a capacidade máxima de tráfego de dados entre o UE e a rede 5G, passando pelo componente do núcleo UPF. Nesse experimento, também foram coletadas as métricas de uso de processador, memória RAM, disco e rede de todos os componentes do núcleo da rede 5G e do testador, utilizando-se as ferramentas para exportação de métricas dos contêineres *Docker*.

Em relação aos parâmetros usados para a execução do experimento, foi utilizada a configuração de 1, 2, 4, 6, 8 e 10 gNBs com 1 UE cada registrando e conectando o plano de dados antes da execução de cada teste de desempenho. Quanto aos parâmetros da ferramenta *iPerf*, foi definido o tempo de execução do experimento em 60 s, reportando as métricas do teste a cada segundo. Os núcleos de rede 5G testados foram o *free5GC* v3.0.6 e o *Open5GS* v2.3.6, ambas recomendadas pelo testador. Esse experimento foi executado 16 vezes, para obter-se uma quantidade de dados relevantes, onde ruídos externos ao experimento fossem amenizados.

Algumas limitações foram observadas durante a preparação e execução desse experimento. Visto que no experimento anterior foi utilizada a versão mais recente do núcleo *free5GC*, havia a intenção de se utilizar a mesma versão para esse experimento. Porém os tutoriais para a implantação da versão v3.2.1 falharam ao estabelecer as rotas neces-

³<https://iperf.fr/>

sárias para o plano de dados, inviabilizando a conexão entre o cliente e o servidor do *iPerf*. Desse modo, foi decidido utilizar uma versão anterior deste núcleo, que havia sido homologada pelos desenvolvedores do testador. Nesse experimento, o OAI apresentou o mesmo problema, com falhas ao estabelecer as rotas entre o UE e a rede 5G nas versões v1.3.0 e v1.4.0, sendo esta a mais recente até o presente momento. Dessa forma, foi decidido não realizar os experimentos sobre esse núcleo.

5.2 Análise e discussão dos tempos de registro e estabelecimento de sessão do núcleo 5G

Para o experimento de medição do tempo de conexão de UEs, foi levado em consideração o tempo total entre o início do processo de registro e a obtenção do plano de dados. Após a coleta e processamento dos dados desse experimento, os gráficos da Figura 5.1 foram gerados para permitir a visualização e análise dos dados coletados. O gráfico representa o tempo médio de conexão para cada rajada de conexões simultâneas, com o eixo X representando o tempo de execução do experimento, em s, e o eixo Y representando o tempo total para estabelecer a conexão e iniciar o plano de dados deste UE, em ms. Na coluna da esquerda, são apresentados os resultados para o núcleo 5G *free5GC* e, na direita, o *Open5GS*. Para cada núcleo, foram desenhados 5 gráficos, com as diferentes variações de intervalo entre rajadas de conexões de UEs.

Ao analisar os dados, é possível inferir que, para as condições do experimento, o núcleo *free5GC* possuiu tempo de conexão médio de 1089 ms, quando cada conexão se iniciou 500 ms após a anterior. Nesse experimento, a mediana do tempo de conexão foi de 853 ms e o tempo máximo de conexão foi de 5,57 s. Para um intervalo entre conexões de 100 ms, o tempo médio de conexão foi de 7731 ms, tendo sua mediana em 3027 ms e o tempo máximo de conexão durando pouco mais de 54 s.

Quanto ao comportamento do núcleo de rede 5G *Open5GS*, percebe-se que para um intervalo de 500 ms, o tempo médio foi de 306 ms, com sua mediana em 305 ms e seu tempo máximo de conexão em 345 ms. Para o intervalo de 100 ms, a média subiu para 364 ms, enquanto a mediana subiu para 358 ms e o valor máximo atingiu 490 ms. Desse modo, considerando-se apenas o tempo de conexão, o núcleo de rede *Open5GS* possui melhor desempenho, aguentando diversas rajadas de conexões em diferentes cenários, tendo seu tempo médio variando em torno de 58 ms. O núcleo *free5GC* possui uma variação de aproximadamente 49 s entre as diferentes execuções do experimento. Os

dados detalhados para cada execução desse experimento podem ser vistos na Tabela 5.2.

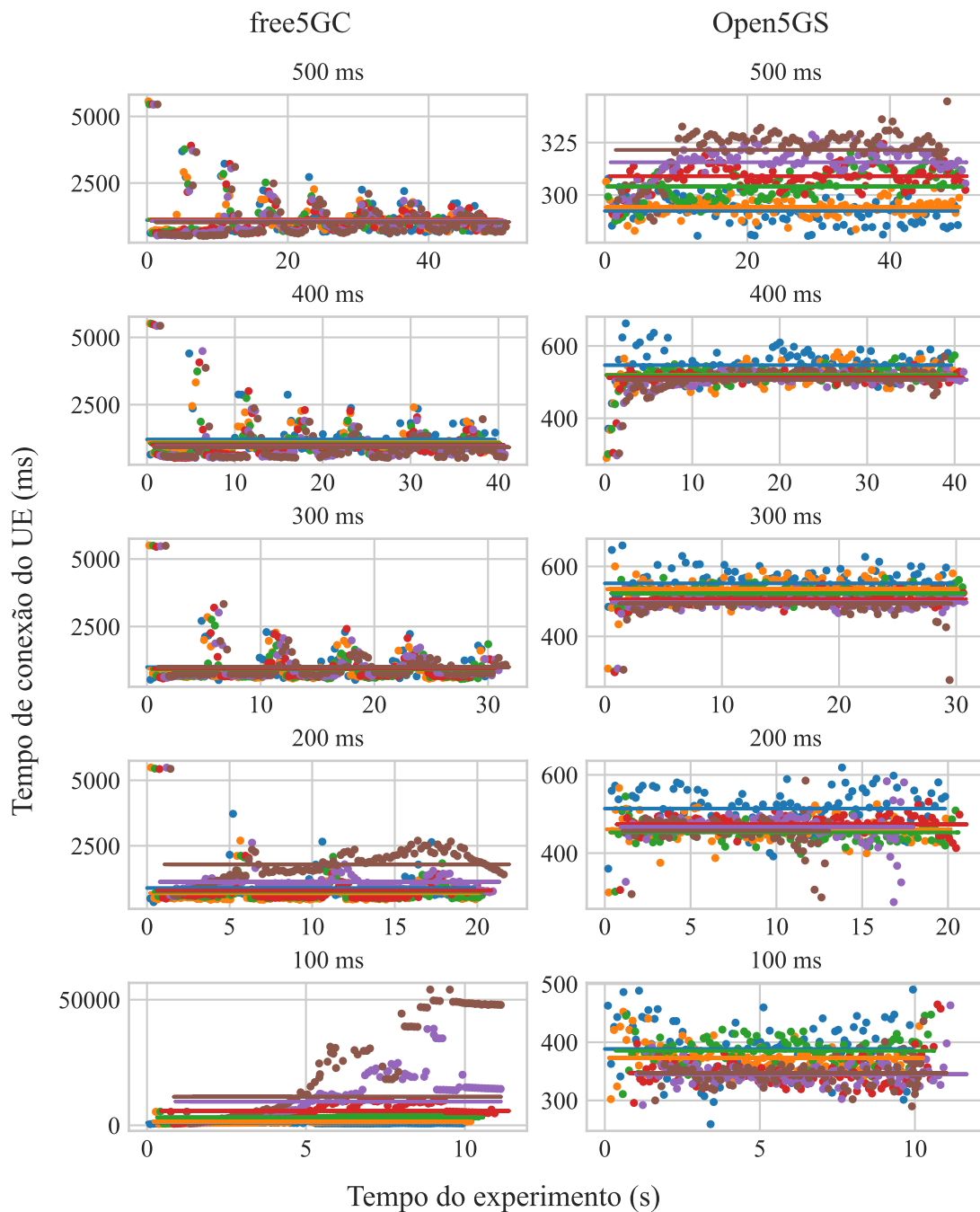


Figura 5.1 – Tempo de registro e estabelecimento de sessão PDU para cada núcleo 5G

Durante a execução dos experimentos, a taxa de conexões mal sucedidas entre o UE e o núcleo da rede foi significativa entre cada execução do experimento para cada um dos núcleos testados. Sendo assim, foram desenhados os gráficos da Figura 5.2 com a comparação de taxa de erro médio entre os dois núcleos avaliados. Cada gráfico representa a taxa de erro médio de conexão dos UEs com os núcleos testados, com o intervalo de tempo entre as rajadas em ordem decrescente, de 500 a 100 ms, com passos de 100

ms. As barras dos gráficos representam a média de erros entre as execuções do experimento realizadas para cada configuração de gNB e núcleo. As linhas representam o desvio padrão da média para cada execução do experimento.

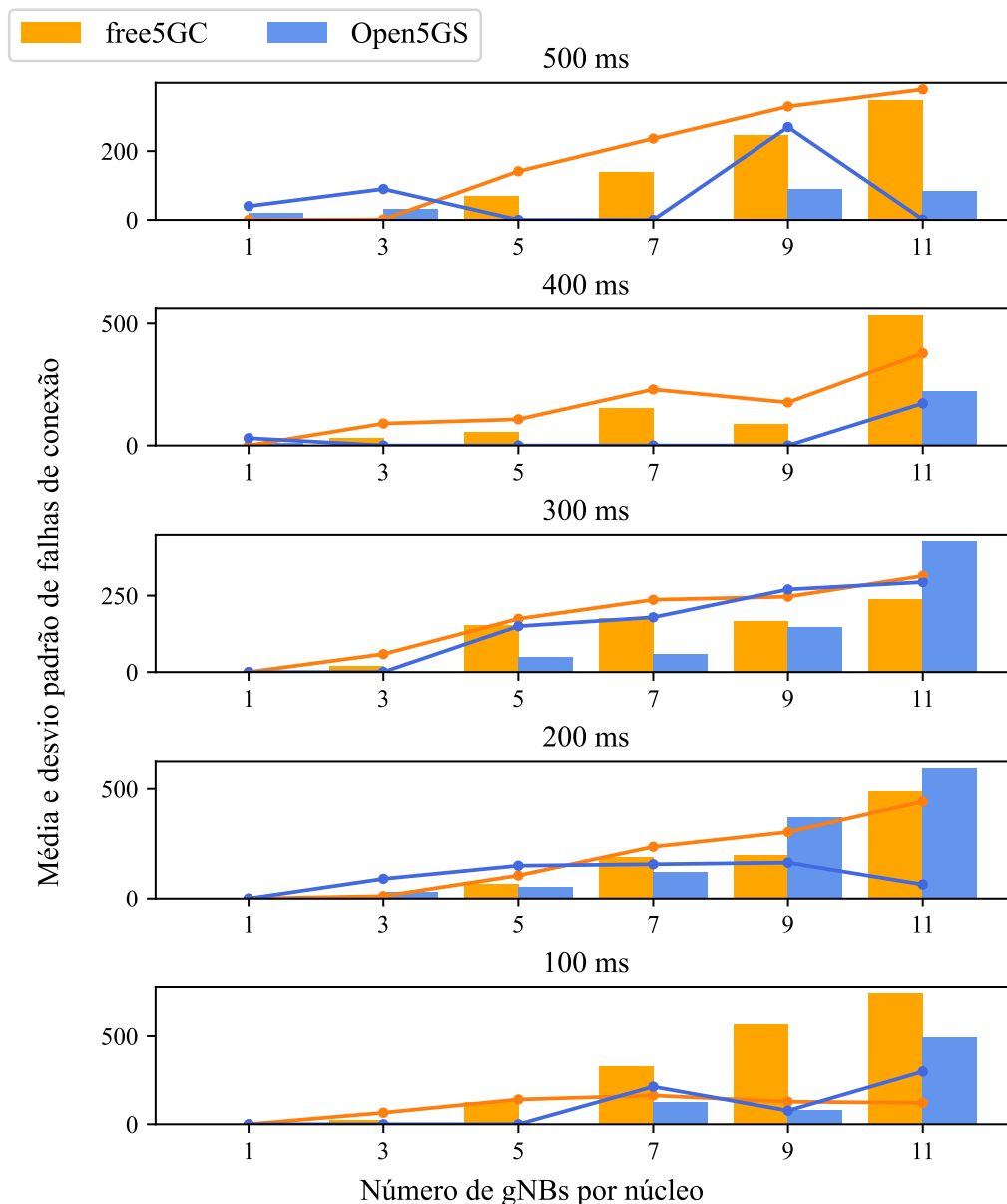


Figura 5.2 – Média e desvio padrão dos erros de conectividade

Nesse experimento, foi possível observar que o núcleo *free5GC* possui uma taxa de erros de conexão crescente para cada gNB adicionada. Ao adicionar uma gNB, a taxa de conexões simultâneas aumentava na mesma proporção que a taxa de erros. Esse comportamento pode ser percebido no núcleo *Open5GS* quando se reduz o intervalo entre rajadas de conexões para um valor menor que o tempo médio necessário para se estabelecer a conexão. Entretanto, para intervalos maiores de tempo, o núcleo *Open5GS* apresentou

Tabela 5.2 – Tabela detalhando os resultados dos experimentos

Núcleo 5G	Atraso entre conexões	Tempo máximo de estabelecimento da sessão PDU	Média de tempo de estabelecimento da sessão PDU	Mediana de tempo de estabelecimento da sessão PDU	Desvio padrão da média de tempo de estabelecimento da sessão PDU
<i>free5GC</i>	500 ms	5566,66 ms	1089,22 ms	853,51 ms	668,45 ms
<i>free5GC</i>	400 ms	5518,75 ms	1029,41 ms	832,78 ms	664,64 ms
<i>free5GC</i>	300 ms	5502,93 ms	958,55 ms	740,69 ms	591,63 ms
<i>free5GC</i>	200 ms	5489,18 ms	997,05 ms	742,37 ms	679,61 ms
<i>free5GC</i>	100 ms	54029,05 ms	7731,11 ms	3027,71 ms	11440,72 ms
<i>Open5GS</i>	500 ms	344,67 ms	306,02 ms	304,65 ms	12,95 ms
<i>Open5GS</i>	400 ms	662,27 ms	518,87 ms	519,20 ms	36,58 ms
<i>Open5GS</i>	300 ms	659,85 ms	517,89 ms	514,54 ms	38,66 ms
<i>Open5GS</i>	200 ms	618,68 ms	470,99 ms	470,00 ms	43,59 ms
<i>Open5GS</i>	100 ms	490,48 ms	364,45 ms	357,73 ms	34,10 ms

boa estabilidade no geral, tendo em alguns momentos a taxa de erros de conexão igual a zero.

Através da análise desses resultados, percebe-se que o núcleo de rede 5G *Open5GS* possui maior estabilidade para o registro de sessões dos UE em escala em comparação com o núcleo *free5GC*. Apesar desses resultados, foi descoberta uma limitação no núcleo *Open5GS* em relação à quantidade máxima de dispositivos conectados simultaneamente, sendo essa igual a 1024 dispositivos. É importante ressaltar que essa limitação não está presente no núcleo *free5GC*.

5.3 Análise e discussão dos resultados em relação ao desempenho do plano de dados

Para o experimento de avaliação do desempenho do plano de dados do núcleo da rede 5G, foi executada, por 60 s, a ferramenta *iPerf* 2.0 simultaneamente em cada UE conectado ao núcleo 5G. Em um primeiro momento, foi utilizada a máquina virtual em sua configuração máxima, com 12 núcleos virtuais de CPU e 8 GB de memória RAM. Após a coleta e o processamento dos dados gerados pela ferramenta *iPerf*, foram gerados os gráficos representando a largura de banda por segundo dos núcleos de rede 5G *free5GC* e *Open5GS* testados.

A Figura 5.3 apresenta os gráficos com a largura de banda para cada configuração de quantidade de UE para o núcleo *free5GC* na máquina virtual com 12 núcleos virtual de processador e 8 GB de memória RAM.

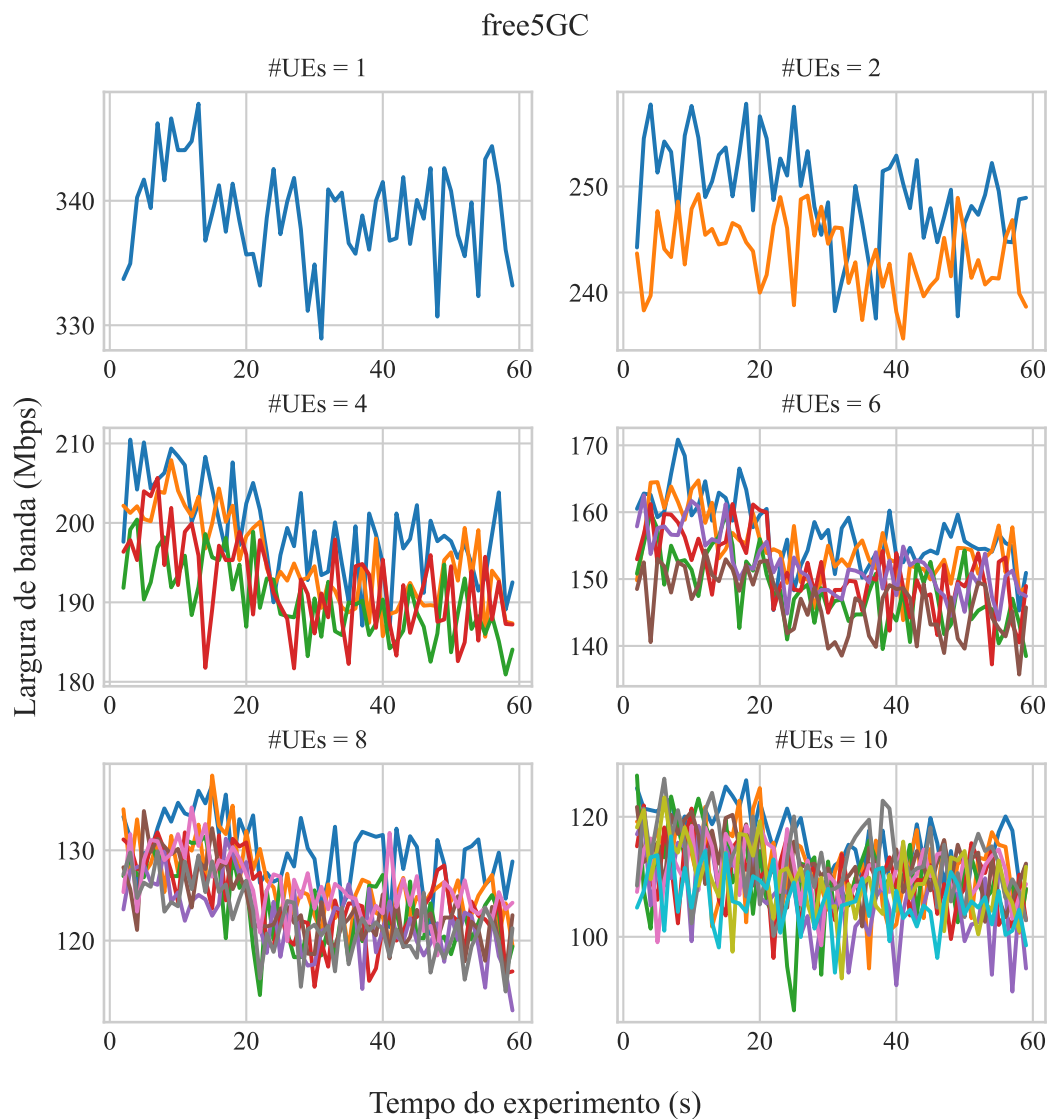


Figura 5.3 – Largura de banda para cada configuração de quantidade de UE para o núcleo *free5GC*

A Figura 5.4 apresenta os gráficos com a largura de banda para cada configuração de quantidade de UE para o núcleo *Open5GS* na máquina virtual com 12 núcleos virtual de processador e 8 GB de memória RAM.

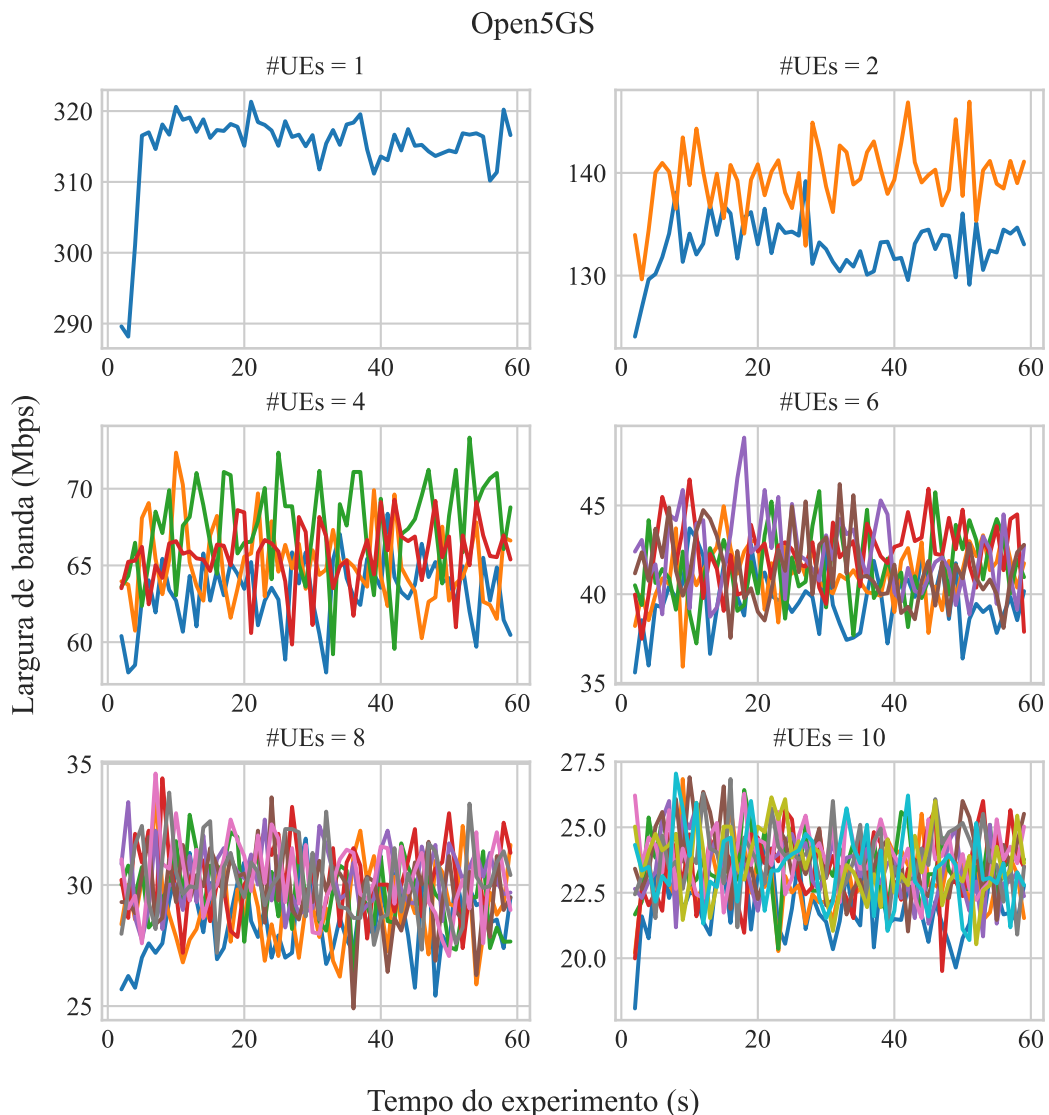


Figura 5.4 – Largura de banda para cada configuração de quantidade de UE para o núcleo *Open5GS*

Através desses gráficos, é possível observar que a largura de banda disponível para os UEs não é estável, oscilando durante toda a execução do experimento. Para uma melhor visualização dos valores agregados de cada execução desse experimento, foram criados gráficos de barra. O eixo X do gráfico representa o número de UEs em cada rodada do experimento, onde segmento da barra representa o UE de mesma cor das duas figuras anteriores. Cada barra do gráfico representa a largura de banda média em Mbps agregada entre os UEs. A Figura 5.5 apresenta o valor agregado da largura de banda para cada configuração de quantidade de UE para a máquina virtual com 12 núcleos virtuais e 8 GB de memória RAM.

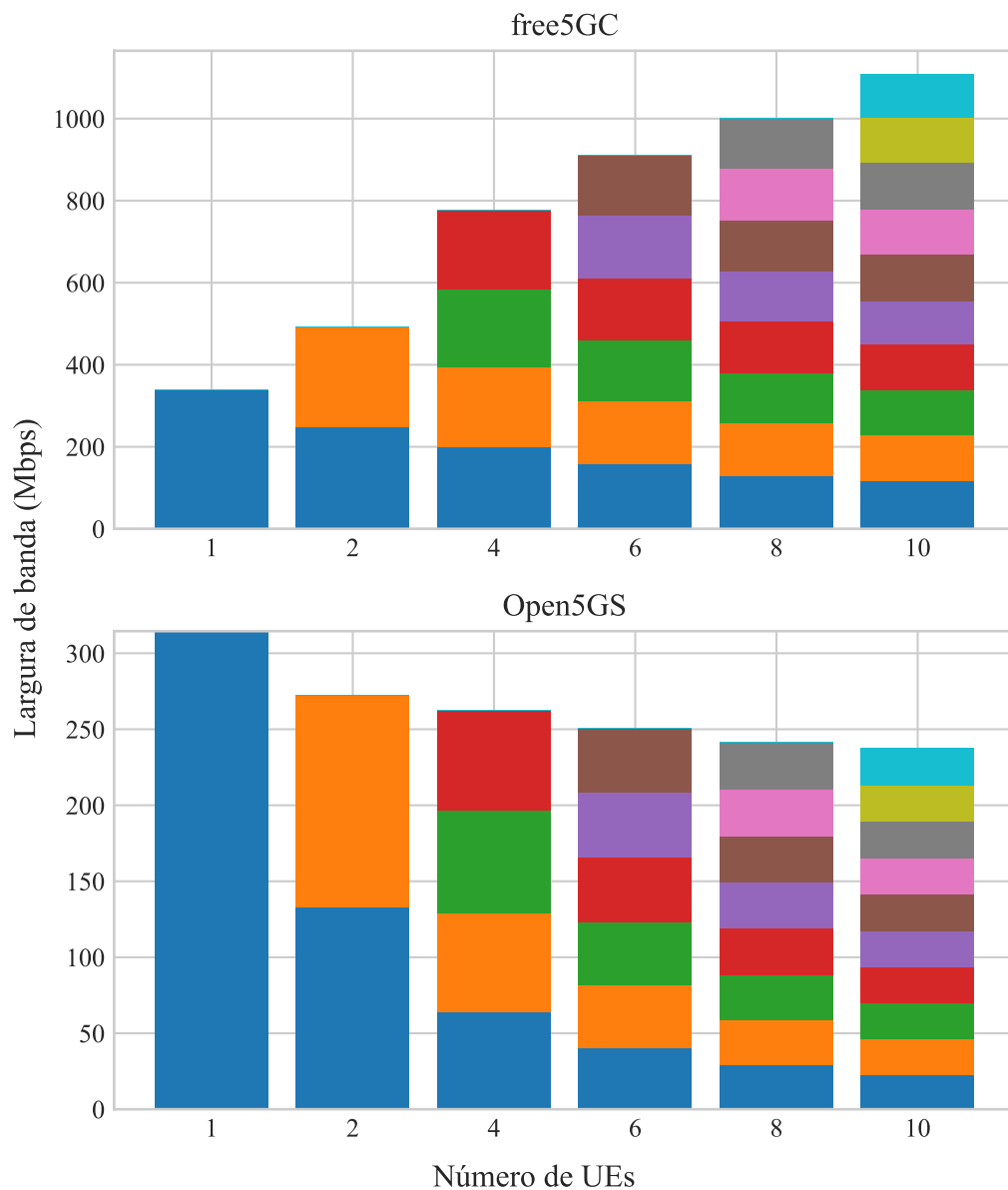


Figura 5.5 – Valor agregado da largura de banda para cada configuração de quantidade de UEs

Nesse experimento, percebe-se que no núcleo *free5GC* a largura de banda total entre cada execução aumenta de acordo com a quantidade de UEs conectados. O valor médio de largura de banda para esse núcleo com um UE conectado é de 338.4 Mbps, aumentando para 777.0 Mbps com quatro UEs conectadas e 1109.9 Mbps com 10 UEs conectadas. Em contrapartida, o comportamento do núcleo *Open5GS* se opõe ao *free5GC*. No núcleo *Open5GS*, o valor médio da largura de banda para um UE conectado é de 314.5 Mbps, reduzindo para 262.7 Mbps com 4 UEs conectadas e 237.9 Mbps ao conectar 10 UEs simultâneas. Esse experimento demonstra que o núcleo *free5GC* possui

melhor estrutura para gerenciar o tráfego de dados de UEs em escala em relação ao núcleo *Open5GS*.

Ao limitar a quantidade de recursos disponíveis para a execução do experimento, diminuindo a quantidade de núcleos virtuais de processador e memória RAM da máquina virtual, percebe-se que o desempenho do plano de dados para o núcleo *free5GC* é diretamente proporcional à quantidade de núcleos virtuais disponíveis. A Figura 5.6 representa o desempenho médio agregado do plano de dados para a execução do experimento em uma máquina virtual com 8 núcleos virtuais de processador e 8 GB de memória RAM. Nesse caso, largura de banda máxima do plano de dados para o núcleo *free5GC* é reduzida para 1004.2 Mbps quando executado com dez UEs simultâneos. Essa redução de desempenho pode ser explicada devido à limitação na carga máxima de processamento suportada pela máquina virtual.

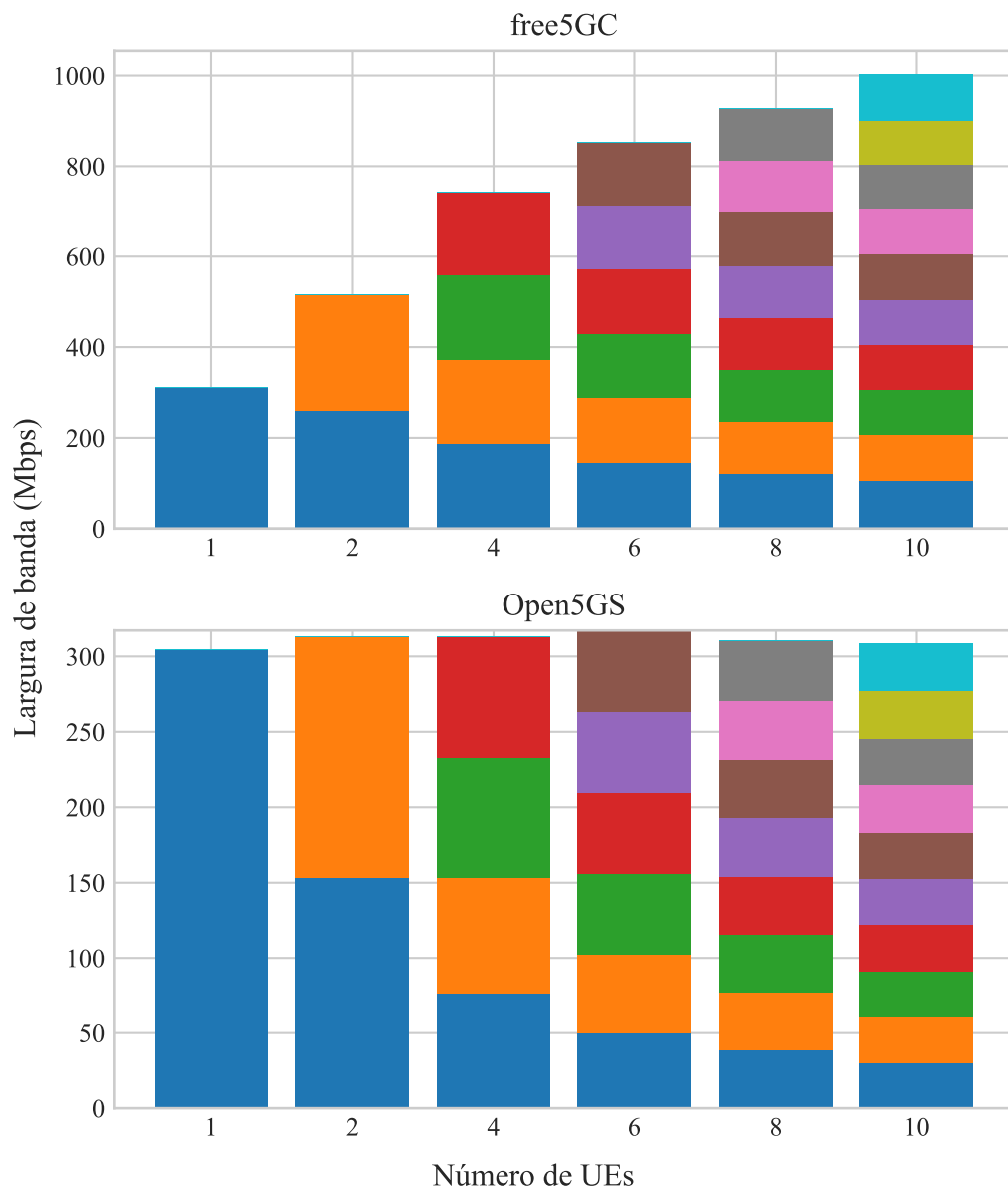


Figura 5.6 – Valor agregado da largura de banda para cada configuração de quantidade de UEs

A redução de desempenho do plano de dados fica mais visível ao limitar-se ainda mais a quantidade de núcleos virtuais disponíveis para o experimento. Na Figura 5.7, onde a máquina virtual é executada com 6 núcleos virtuais de processador e 8 GB de memória RAM, é possível perceber que o desempenho máximo do plano de dados do núcleo *free5GC* é atingido com seis UEs simultâneos.

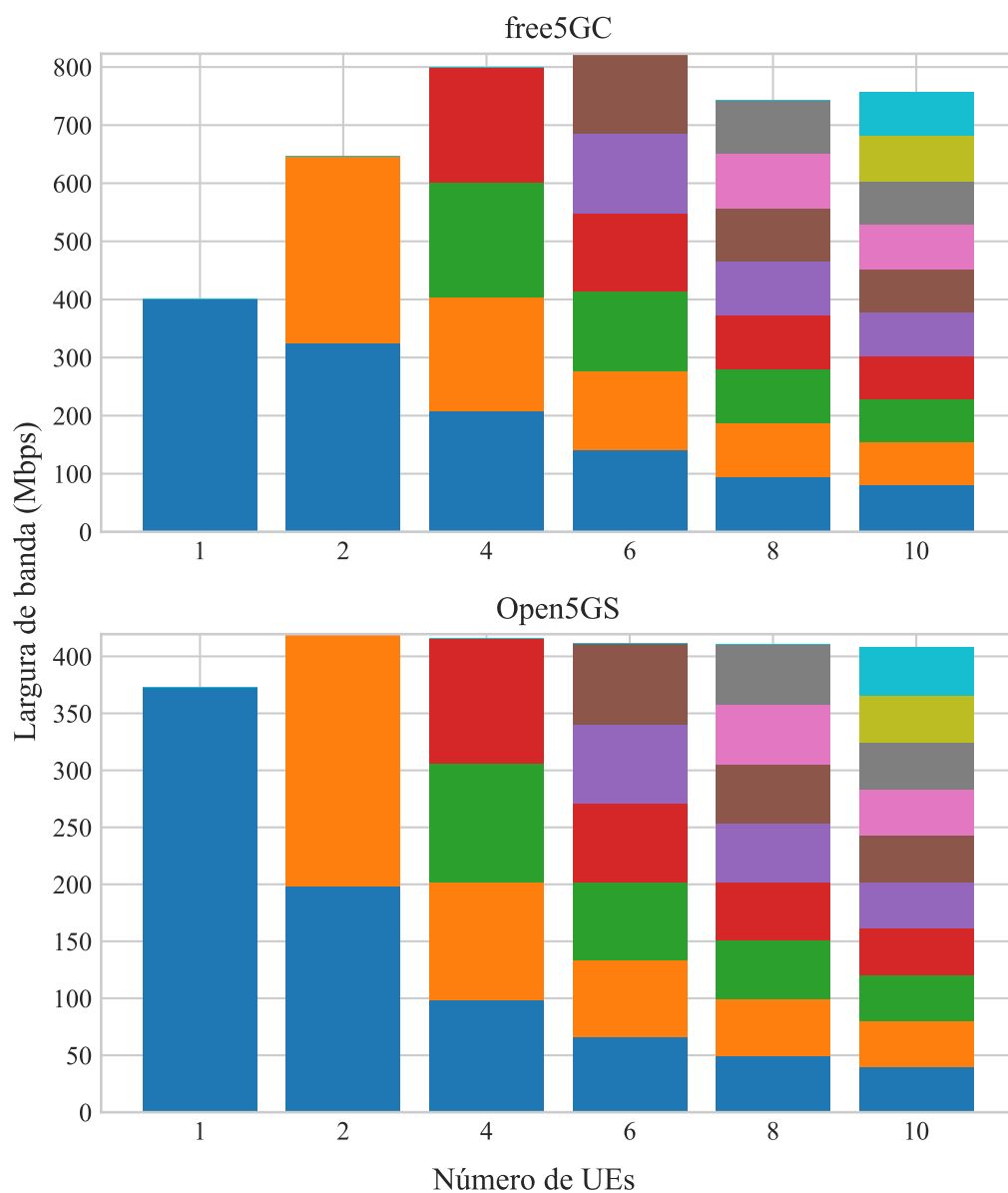


Figura 5.7 – Valor agregado da largura de banda para cada configuração de quantidade de UEs

A Figura 5.8 representa os resultados obtidos para a execução do experimento em uma máquina virtual com 4 núcleos virtuais de processador e 4 GB de memória RAM. Essa configuração foi utilizada para simular um cenário com recursos limitados. Neste caso, o desempenho máximo do núcleo *free5GC* ocorre com quatro UEs simultâneos. A redução de desempenho para testes com maior quantidade de UEs simultâneos é explicada pela sobrecarga de uso de processador causada pela quantidade de UEs trafegando dados em paralelo. Os detalhes de uso de recursos são apresentados na Seção 5.4.

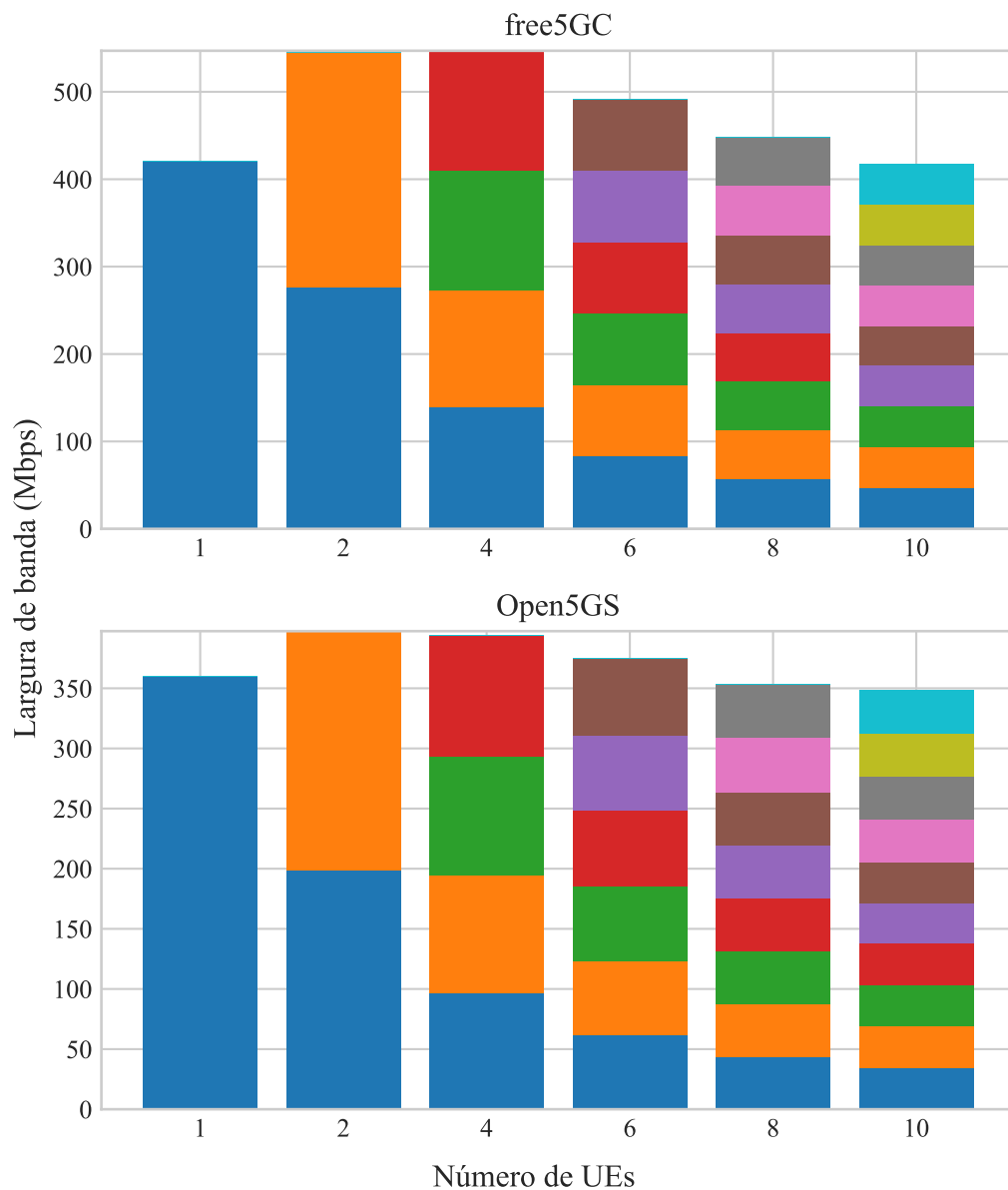


Figura 5.8 – Valor agregado da largura de banda para cada configuração de quantidade de UEs

5.4 Análise e discussão em relação ao consumo de recursos

Esta seção apresenta a análise de dados de uso de processador e memória RAM da máquina virtual para a execução dos experimentos. Os dados para cada contêiner foram agrupados em três conjuntos, para facilitar a visualização dos dados. O primeiro conjunto representa os dados agregados de todas as instâncias do testador, denominado nos gráficos como Testadores. O segundo conjunto representa os dados agregados de todas as

funções do núcleo em teste, denominado nos gráficos como Núcleo. O terceiro conjunto representa os dados agregados do restante dos serviços em execução, como os contêineres utilizados para rodar o módulo de coleta de dados do núcleo e o servidor da aplicação *iPerf* utilizada no segundo experimento, denominado nos gráficos como Outros. Uma análise com maiores detalhes do uso de recursos para cada componente em execução durante o período do experimento é possível ao utilizar a ferramenta para gerar visualizações dos dados coletados disponível através da aplicação *InfluxDB*. Essa ferramenta faz parte do módulo de coleta de dados do núcleo da rede.

5.4.1 Análise e discussão em relação ao consumo de recursos do experimento de registro e estabelecimento de sessão do núcleo 5G

Para a análise do uso de recursos de processador e memória RAM para a execução do experimento de registro e estabelecimento de sessão do núcleo 5G, foi utilizada uma execução do experimento com intervalo entre conexões de UEs de 500 ms sobre a máquina virtual com 12 núcleos de processador virtuais e 8 GB de memória RAM. Os resultados agregados podem ser vistos na Figura 5.9.

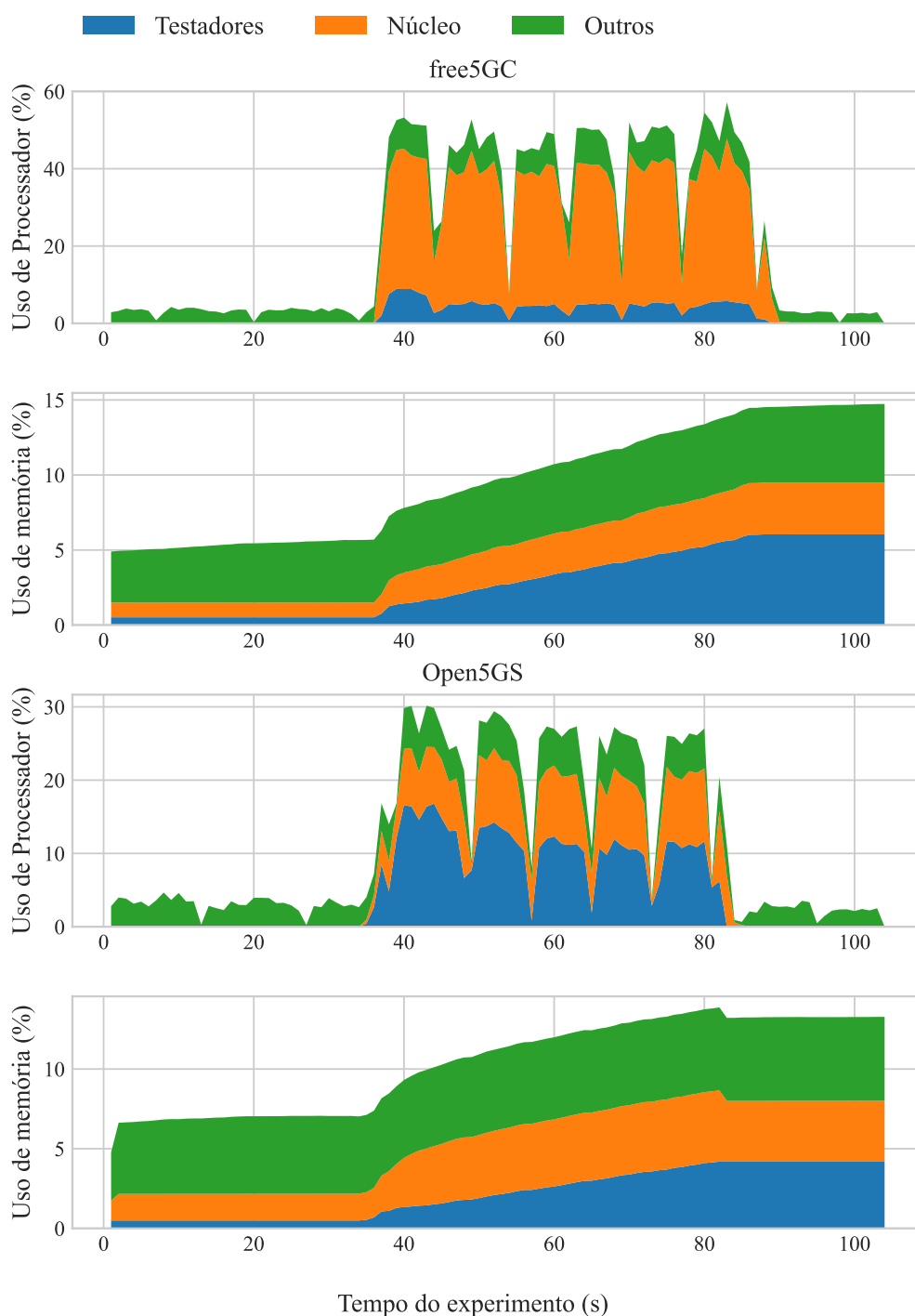


Figura 5.9 – Uso de processador e memória RAM para a execução com atraso de 500 ms entre conexões

Esta figura apresenta o uso de processador e memória RAM de acordo com o tempo de execução do experimento para os núcleos *free5GC* e *Open5GS*. O eixo X dos gráficos representam o tempo em segundos referente à coleta de dados total do experimento, considerando o tempo para a inicialização, execução e encerramento do experimento. O eixo Y representa a porcentagem do uso para cada tipo de recurso analisado.

As áreas azuis, laranjas e verdes dos gráficos representam a porcentagem de uso agregado de recursos para a execução dos testadores, núcleos 5G e os demais componentes, respectivamente.

Ao observar o uso de recursos somente durante o período da execução do experimento, a média de uso de processador para o núcleo *free5GC* foi de 31,33% e para o núcleo *Open5GS* foi de 7,26%. Em relação ao uso de memória RAM, o pico de uso de memória para o núcleo *free5GC* foi de 3,45%, representando o uso de 282,6 MB de memória RAM. Quanto ao núcleo *Open5GS*, o pico de uso de memória RAM foi de 4,47%, representando o uso de 366,2 MB de memória RAM. Os resultados descritos acima demonstram que o núcleo de rede 5G *Open5GS* apresentou uma performance 4.3x superior em relação ao uso de processador em comparação com o núcleo *free5GC*. Entretanto, em relação ao uso de memória RAM, o núcleo *free5GC* apresentou uma performance 1.29x superior ao núcleo *Open5GS*.

5.4.2 Análise e discussão em relação ao consumo de recursos do experimento de desempenho do plano de dados

Para realizar a análise do uso de recursos durante a execução do experimento para a avaliação do desempenho do plano de dados, foi escolhida as métricas de uso de recursos do processador e memória RAM. Essa análise complementa a explicação do experimento demonstrada na Seção 5.3. Os experimentos foram conduzidos em quatro configurações diferentes. Entretanto, a análise do consumo de recursos foi focada em apenas duas configurações, com a maior e com a menor quantidade de recursos. Ao observar os resultados do plano de dados para a execução do experimento de desempenho do plano de dados na máquina virtual com 12 núcleos virtuais e 8 GB de memória RAM, percebe-se que o desempenho máximo para o plano de dados do núcleo *free5GC* com essa configuração de máquina virtual não foi atingido. Esse resultado pode ser visto nos gráficos da Figura 5.10.

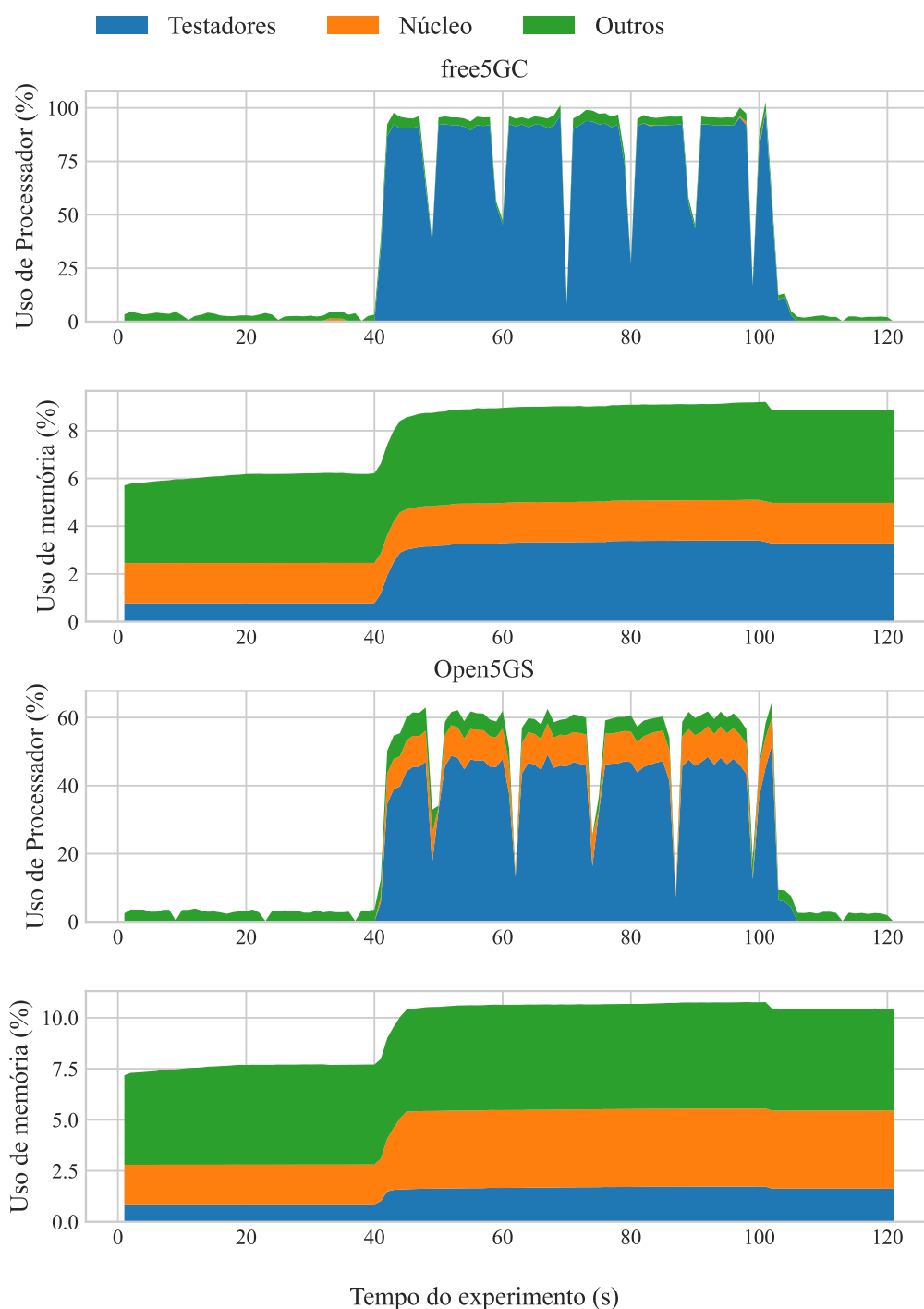


Figura 5.10 – Uso de processador e memória RAM para o experimento de desempenho de plano de dados com 10 UEs conectados

É possível observar que o uso médio de processador durante a execução desse experimento foi de 95,57% para o núcleo *free5GC*, com poucos picos de uso atingindo os 100%. Do total de uso de processador, 91,87% do uso é representado pelos testadores e UEs realizando os testes. O uso de processador médio do núcleo *free5GC* durante a execução do experimento foi de 0,03%. Isso demonstra que o núcleo de rede 5G utiliza

poucos recursos de processador para realizar o gerenciamento do plano de dados dos UEs conectados.

Em relação ao núcleo *Open5GS*, que não é otimizado para o gerenciamento do plano de dados de múltiplos UEs em paralelo, o uso de processador médio total durante a execução do experimento foi de 59,59%. O total de uso de processador durante a execução de experimento é dividida principalmente entre as instâncias do testador e o núcleo da rede 5G, com o testador representando 45,95% do uso do processador e o núcleo representando 8,96%. A redução de uso de processador pelas instâncias do testador e UEs em relação ao experimento executado com o núcleo *free5GC* é explicada pela baixa largura de banda disponível pelo *Open5GS*.

Uma possível explicação desses resultados é que o núcleo *free5GC* provavelmente utiliza paralelização para gerenciar o processamento do plano de dados dos UEs. Isso provavelmente justificaria a facilidade em escalar a quantidade de UEs conectados reduzindo sutilmente o desempenho individual dos UEs. Por outro lado, o núcleo *Open5GS* não apresenta o mesmo comportamento. De acordo com os resultados, o núcleo utilizou 8,96% do total de processamento disponível na máquina virtual. Isso é equivalente a um pouco mais que a capacidade de um núcleo do processador disponível. Possivelmente, este núcleo não está otimizado para paralelizar o processamento do plano de dados quando múltiplos UEs estão conectados. Isso explicaria o baixo desempenho observado nos resultados do núcleo *Open5GS*.

Ao observar o uso de recursos de processador para a máquina virtual com 4 núcleos virtuais de processador e 4 GB de memória RAM, é possível explicar a queda de desempenho para o plano de dados do núcleo *free5GC* após os testes com mais de 4 UEs simultâneos. A Figura 5.11 representa os gráficos de uso de processador e memória RAM para uma execução do experimento de desempenho do plano de dados

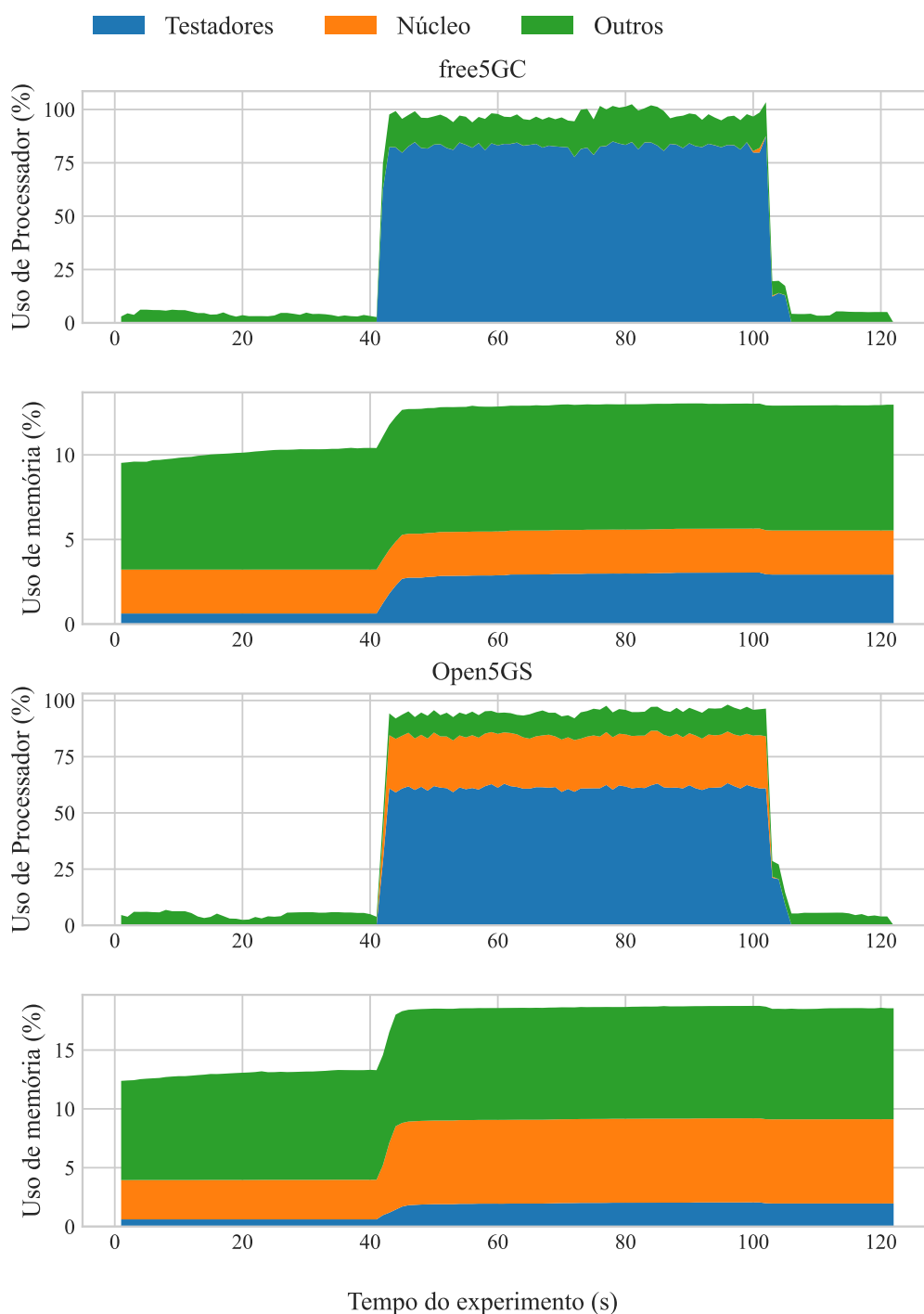


Figura 5.11 – Uso de processador e memória RAM para o experimento de desempenho de plano de dados com 4 UEs conectados

É possível observar que o uso do processador para a execução de quatro UEs conectados e utilizando o plano de dados do núcleo de rede 5G *free5GC* está muito próximo de 100%, atingindo o uso máximo do processador em diversos momentos. Dessa forma, é possível perceber que o máximo de UEs simultâneos que essa configuração de máquina virtual suporta são quatro. Para quantidades superiores a quatro UEs simultâneos, a re-

dução de desempenho do plano de dados é explicada devido ao chaveamento necessário realizado pelo sistema operacional para gerenciar os diversos processos executando em paralelo.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O presente trabalho atinge a proposta inicial, desenvolvendo uma ferramenta para testes de desempenho de redes móveis 5G de código aberto, além de avaliar o desempenho em dois distintos cenários com duas implementações de núcleo 5G existentes. As descobertas e limitações encontradas podem ser úteis para o desenvolvimento dos núcleos. Da mesma forma, essa ferramenta pode ser utilizada para avaliar outros núcleos não testados no presente trabalho.

Este trabalho apresenta duas principais contribuições teóricas. A primeira é com relação aos testes de desempenho, principalmente sobre redes móveis. A implementação do módulo de execução dos testes de desempenho e análise dos dados coletado traz discussões relevantes sobre a utilidade do uso de testes de desempenho durante o desenvolvimento de software. Essas discussões são de extrema relevância para a área de Engenharia de Software, mostrando que os testes de desempenho ajudam a validar implementações de software para operarem de maneira correta sobre alta carga de trabalho.

A segunda contribuição teórica deste trabalho é em relação à estabilidade das atuais implementações de código aberto de núcleos de rede 5G. As três implementações avaliadas possuem grandes limitações em relação ao desempenho em escala, tornando sua implantação em redes reais inviável neste momento. Esse trabalho demonstra que a implementação OAI, sendo a menos madura delas, não conseguiu rodar nenhum dos testes de desempenho devido a suas limitações. Por outro lado, as outras duas implementações avaliadas apresentam diversas limitações que fizeram com que o módulo de testes de desempenho desenvolvido fosse adaptado para funcionar sobre as limitações encontradas e descritas no decorrer do trabalho.

Este trabalho também possui contribuições práticas, ao desenvolver um módulo para ser acoplado no testador *my5G-RANTester* de núcleos 5G. Este módulo está disponível no repositório de código do projeto PORVIR-5G e é aberto para a comunidade. Dessa forma, é possível utilizar a implementação do testador disponível para realizar testes sobre novas versões das implementações existentes de núcleos 5G e para outras implementações futuras.

Este trabalho possui algumas limitações. Dentre elas, pode-se citar as limitações dos núcleos testados, fazendo com que os testes em escala fossem limitados a uma menor quantidade de UEs se registrando, estabelecendo uma sessão PDU e trafegando dados. Outra limitação relevante é em relação à execução do testador na mesma máquina virtual

onde o núcleo foi executado. Ao executar todo o ambiente de testes na mesma máquina que está sendo executado o núcleo da rede 5G, os recursos de hardware disponíveis são divididos entre os processos em execução. Dessa forma, o uso de memória RAM e processador pelo testador para gerenciar as diversas conexões de UEs e pelos UEs para executar testes de largura de banda reduz a quantidade de recursos disponíveis para o núcleo, afetando o seu desempenho.

O presente trabalho abre oportunidades para futuros experimentos em relação ao desempenho de núcleos de redes 5G. Um exemplo seria o teste sobre outras implementações de núcleos de redes 5G de código aberto não avaliadas no presente trabalho. Outro tópico para pesquisas seria avaliar o desempenho de implementações comerciais de núcleos de redes 5G. Além disso, pode-se utilizar a implementação atual do testador, executando testes sobre ambientes de fácil escalabilidade, permitindo executar o testador em uma máquina virtual separada da máquina que está executando o núcleo a ser testado. Dessa forma, o uso de recursos do testador não deve afetar a disponibilidade de recursos para o núcleo, tendo resultados mais relevantes em relação ao funcionamento da implementação em teste.

REFERÊNCIAS

- 3GPP. *Technical Specifications and Technical Reports for a GERAN-based 3GPP system*. [S.l.], 2009. Version 8.20.0. Available from Internet: <<http://www.3gpp.org/DynaReport/0101.htm>>.
- 3GPP. *Technical Specifications and Technical Reports for a 5G based 3GPP system*. [S.l.], 2020. Version 16.0.0. Available from Internet: <<http://www.3gpp.org/DynaReport/21205.htm>>.
- 3GPP. *Architecture enhancements for control and user plane separation of EPC nodes*. [S.l.], 2021. Version 17.0.0. Available from Internet: <<http://www.3gpp.org/DynaReport/23214.htm>>.
- 3GPP. *NG-RAN; NG Application Protocol (NGAP)*. [S.l.], 2021. Version 16.8.0. Available from Internet: <<http://www.3gpp.org/DynaReport/38413.htm>>.
- 3GPP. *NG-RAN; PDU session user plane protocol*. [S.l.], 2021. Version 16.6.0. Available from Internet: <<http://www.3gpp.org/DynaReport/38415.htm>>.
- 3GPP. *5G System; Authentication Server Services; Stage 3*. [S.l.], 2022. Version 17.5.0. Available from Internet: <<http://www.3gpp.org/DynaReport/29509.htm>>.
- 3GPP. *5G System Enhancements for Edge Computing; Stage 2*. [S.l.], 2022. Version 17.2.0. Available from Internet: <<http://www.3gpp.org/DynaReport/23548.htm>>.
- 3GPP. *Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3*. [S.l.], 2022. Version 17.6.0. Available from Internet: <<http://www.3gpp.org/DynaReport/24501.htm>>.
- 3GPP. *System architecture for the 5G System (5GS)*. [S.l.], 2022. Version 17.4.0. Available from Internet: <<http://www.3gpp.org/DynaReport/23501.htm>>.
- 3GPP. *Technical Specifications and Technical Reports for an Evolved Packet System (EPS) based 3GPP system*. [S.l.], 2022. Version 17.0.0. Available from Internet: <<http://www.3gpp.org/DynaReport/21201.htm>>.
- AHMAD, I. et al. Security for 5g and beyond. **IEEE Communications Surveys & Tutorials**, Institute of Electrical and Electronics Engineers Inc., v. 21, p. 3682–3722, 10 2019. ISSN 1553-877X. Available from Internet: <<https://ieeexplore.ieee.org/document/8712553/>>.
- BERTOLINO, A. Software testing research and practice. In: _____. [s.n.], 2003. v. 2589, p. 1–21. Available from Internet: <http://link.springer.com/10.1007/3-540-36498-6_1>.
- BHADRA, D. R. et al. Packet loss probability in wireless networks: A survey. In: . IEEE, 2015. p. 1348–1354. ISBN 978-1-4799-8081-9. Available from Internet: <<http://ieeexplore.ieee.org/document/7322729/>>.
- BHALLA, M. R.; BHALLA, A. V. et al. Generations of mobile wireless technology: A survey. **International Journal of Computer Applications**, Citeseer, v. 5, n. 4, p. 26–32, 2010.

BOANO, C. A. et al. Iotbench: Towards a benchmark for low-power wireless networking. In: . IEEE, 2018. p. 36–41. ISBN 978-1-5386-6742-2. Available from Internet: <<https://ieeexplore.ieee.org/document/8429500/>>.

CARDOSO, K. V. et al. A softwarized perspective of the 5g networks. 6 2020. Available from Internet: <<http://arxiv.org/abs/2006.10409>>.

GARCIA, J.; ALFREDSSON, S.; BRUNSTROM, A. Delay metrics and delay characteristics: A study of four swedish hsdpa+ and lte networks. In: . IEEE, 2015. p. 234–238. ISBN 978-1-4673-7359-3. Available from Internet: <<http://ieeexplore.ieee.org/document/7194075/>>.

LAMBA, A.; YADAV, J.; DEVI, G. U. Analysis of technologies in 3g and 3.5g mobile networks. In: . IEEE, 2012. p. 330–333. ISBN 978-1-4673-1538-8. Available from Internet: <<http://ieeexplore.ieee.org/document/6200694/>>.

LEE, K.-L.; LEE, C.-N.; LEE, M.-F. Realizing 5g network slicing provisioning with open source software. In: IEEE. **2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)**. [S.l.], 2021. p. 1923–1930.

MAHN-SUK, Y. et al. Performance test of lte-r railway wireless communication at high-speed (350 km/h) environments. In: . IEEE, 2018. p. 637–640. ISBN 978-1-5386-4646-5. Available from Internet: <<https://ieeexplore.ieee.org/document/8437021/>>.

MIAH, A.; TAN, K. An overview of 3g mobile network infrastructure. In: . IEEE, 2002. p. 228–232. ISBN 0-7803-7565-3. Available from Internet: <<http://ieeexplore.ieee.org/document/1033099/>>.

PATTERSON, D. A.; HENNESSY, J. L. **Computer organization and design**. 3. ed. [S.l.]: Morgan Kaufmann, 2014. (Morgan Kaufmann Series in Computer Architecture and Design G).

PERAL-ROSADO, J. A. del et al. Survey of cellular mobile radio localization methods: From 1g to 5g. **IEEE Communications Surveys & Tutorials**, Institute of Electrical and Electronics Engineers Inc., v. 20, p. 1124–1148, 4 2018. ISSN 1553-877X. Available from Internet: <<https://ieeexplore.ieee.org/document/8226757/>>.

RADATZ, J.; GERACI, A.; KATKI, F. Ieee standard glossary of software engineering terminology. **IEEE Std 610.12-1990**, p. 1–84, 1990.

RAYNER, D. Osi conformance testing. **Computer Networks and ISDN Systems**, v. 14, p. 79–98, 1 1987. ISSN 01697552. Available from Internet: <<https://linkinghub.elsevier.com/retrieve/pii/0169755287900870>>.

REDANA, S. et al. 5g ppp architecture working group view on 5g architecture. 2020. Available from Internet: <<http://doi.org/10.5281/zenodo.3265031>>.

ROJKO, A. Industry 4.0 concept: Background and overview. **International Journal of Interactive Mobile Technologies (iJIM)**, International Association of Online Engineering, v. 11, p. 77, 7 2017. ISSN 1865-7923. Available from Internet: <<http://online-journals.org/index.php/i-jim/article/view/7072>>.

SARIKAYA, B. Conformance testing: Architectures and test sequences. **Computer Networks and ISDN Systems**, v. 17, p. 111–126, 7 1989. ISSN 01697552. Available from Internet: <<https://linkinghub.elsevier.com/retrieve/pii/0169755289900044>>.

SEBASTIAN, S. et al. A survey of various load balancing algorithms in cloud computing innovative research publications review of hybrid load balancing algorithms in cloud computing environment a survey of various load balancing algorithms in cloud computing. **INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH**, v. 3, 2014. ISSN 2277-8616. Available from Internet: <www.ijstr.org>.

SILVEIRA, L. B. et al. Tutorial on communication between access networks and the 5g core. **Computer Networks**, v. 216, p. 109301, 2022. ISSN 1389-1286. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S1389128622003528>>.

TAYYAB, M.; GELABERT, X.; JANTTI, R. A survey on handover management: From lte to nr. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 118907–118930, 2019. ISSN 2169-3536. Available from Internet: <<https://ieeexplore.ieee.org/document/8812724/>>.

VORA, L. J. Evolution of mobile generation technology: 1g to 5g and review of upcoming wireless technology 5g. **International journal of modern trends in engineering and research**, v. 2, n. 10, p. 281–290, 2015.

YI, S. et al. Overview of lte and lte-advanced new features. In: _____. **Radio Protocols for LTE and LTE-Advanced**. John Wiley & Sons, Ltd, 2012. chp. 7, p. 151–158. ISBN 9781118188545. Available from Internet: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118188545.ch7>>.

ZHANG, P. et al. A survey of testing for 5g: Solutions, opportunities, and challenges. **China Communications**, v. 16, p. 69–85, 2019.