



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
DEPARTAMENTO DE ENG. DE CONTROLE, AUTOMAÇÃO E COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Lucas Kienen Rota

**Avaliação e validação da interoperabilidade das Redes Non3GPP com as
redes 5G fim a fim (UE a UE)**

Blumenau
2023

Lucas Kienen Rota

**Avaliação e validação da interoperabilidade das Redes Non3GPP com as
redes 5G fim a fim (UE a UE)**

Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico, de Ciências Exatas e Educação da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Orientador, Dr. Adão Boava

Coorientador: Me. Christian Mailer

Blumenau

2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rota, Lucas Kienen

Avaliação e validação da interoperabilidade das Redes
Non3GPP com as redes 5G fim a fim (UE a UE) / Lucas Kienen
Rota ; orientador, Adão Boava, coorientador, Christian
Mailer, 2023.

88 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Blumenau,
Graduação em Engenharia de Controle e Automação, Blumenau,
2023.

Inclui referências.

1. Engenharia de Controle e Automação. 2. 5G. 3.
NON3GPP. 4. Aether. 5. IIot. I. Boava, Adão. II. Mailer,
Christian. III. Universidade Federal de Santa Catarina.
Graduação em Engenharia de Controle e Automação. IV. Título.

Lucas Kienen Rota

**Avaliação e validação da interoperabilidade das Redes Non3GPP com as
redes 5G fim a fim (UE a UE)**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Engenheiro de Controle e Automação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação.

Blumenau, 05 de 07 de 2023.

Banca Examinadora:

Prof. Dr. Adão Boava
Universidade Federal de Santa Catarina

Prof. Dr. Ciro André Pitz
Universidade Federal de Santa Catarina

Prof. Dr. Maiquel de Brito
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos pilares da minha vida: meus amados pais, minha querida esposa e minha estimada irmã.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão à minha mãe, Jane, e ao meu pai, Didmar, que sempre me apoiaram e me proporcionaram todas as oportunidades para atingir minhas metas. À minha esposa, Maria Elisa, cujo incentivo constante foi uma fonte inabalável de força durante minha jornada e a minha querida irmã Beatriz.

Um agradecimento especial ao meu orientador, Prof. Dr. Adão Boava, e ao meu coorientador, Me. Christian Mailer, cuja orientação valiosa, paciência e vasto conhecimento foram fundamentais. Sua dedicação em compartilhar suas sabedorias e fornecer orientação meticulosa tornaram possível a realização deste trabalho.

RESUMO

Neste trabalho, o objetivo foi estabelecer e comparar a funcionalidade e interoperabilidade das conexões NON3GPP (*NON Third Generation Partnership Project*) e 3GPP (*Third Generation Partnership Project*) com o núcleo de rede 5G. Utilizando os ambientes de simulação Aether e UERANSIM, foram realizados experimentos e análises de desempenho para avaliar a eficiência e qualidade dessas conexões. O Aether foi utilizado como o núcleo do 5G, enquanto o UERANSIM emulou dispositivos de usuário e RAN (*Radio Access Network*) (gNodeB) estabelecendo uma conexão 3GPP com o núcleo do 5G, enquanto foi utilizado um túnel NWu para estabelecer uma conexão NON3GPP. Os resultados obtidos contribuem para um melhor entendimento da integração de conexões NON3GPP em redes 5G, especialmente em ambientes industriais e de IoT (*Internet of Things*), onde a presença de dispositivos NON3GPP é essencial. Foi possível validar a conexão entre os UE (*User Equipment*)s 3GPP e NON3GPP, além de coletar dados de Qualidade de Serviço (QoS), uso de CPU e memória RAM para cada sistema. Comprovou-se que, embora viável e segura, a conexão NON3GPP apresenta uma qualidade de serviço inferior e requer mais recursos computacionais.

Palavras-chave: 5G; NON3GPP; Aether; IIoT; Interoperabilidade.

ABSTRACT

In this study, the aim was to establish and compare the functionality and interoperability of NON3GPP (*NON Third Generation Partnership Project*) and 3GPP (*Third Generation Partnership Project*) connections with a 5G network core. Using the Aether and UERANSIM simulation environments, experiments and performance analyses were conducted to evaluate the efficiency and quality of these connections. Aether was used as the 5G core, while UERANSIM emulated user devices and RAN (*Radio Access Network*) (gNodeB), establishing a 3GPP connection with the 5G core, while a NWu tunnel was used to establish a NON3GPP connection. The results obtained contribute to a better understanding of the integration of NON3GPP connections into 5G networks, especially in industrial and IoT (*Internet of Things*) settings, where the presence of NON3GPP devices is crucial. It was possible to validate the connection between the UE (*User Equipment*)s 3GPP and NON3GPP, and to collect Quality of Service (QoS), CPU usage, and RAM data for each system. We confirmed that, although viable and secure, the NON3GPP connection presents a lower service quality and requires more computational resources.

Keywords: 5G; NON3GPP; Aether; IIoT; Interoperability.

LISTA DE FIGURAS

Figura 1 – Evolução arquitetural das redes móveis.	19
Figura 2 – Arquitetura de um sistema de telecomunicações móveis.	21
Figura 3 – Arquitetura de acesso para rede NON3GPP.	22
Figura 4 – Sistema Nwu-Aether.	24
Figura 5 – Categorias de serviço.	25
Figura 6 – Funções de rede essenciais do núcleo 5G.	27
Figura 7 – Interfaces de rede utilizadas pelo AMF.	28
Figura 8 – Interfaces de rede utilizadas pelo SMF.	29
Figura 9 – Interfaces de rede utilizadas pelo UPF.	30
Figura 10 – Interfaces de rede utilizadas pelo NSSF.	33
Figura 11 – Pilha de protocolos da RAN 5G.	34
Figura 12 – Feixe único e multi-feixe.	35
Figura 13 – UM-MIMO.	36
Figura 14 – SU-MIMO.	36
Figura 15 – Fatiamento de Rede.	37
Figura 16 – Sistema UERANSIM-Aether.	40
Figura 17 – Sistema completo.	42
Figura 18 – Conexão entre UERANSIM e Aether.	51
Figura 19 – Gráfico de <i>throughput</i>	65
Figura 20 – Gráfico de <i>throughput</i> sem UE-gNB para o Aether e VM-Direta.	66
Figura 21 – Gráfico de latência.	66

LISTA DE TABELAS

Tabela 1 – Requisitos de serviço para o 5G.	26
Tabela 2 – Resultados obtidos. $U_N = UE-NON3GPP$ e $U_g = UE-gNB$	67

LISTA DE ABREVIATURAS E SIGLAS

3GPP	<i>Third Generation Partnership Project</i>
5G	<i>Fifth Generation</i>
5G-AN	<i>5G Access Network</i>
AMF	<i>Access and Mobility Management Function</i>
AUSF	<i>Authentication Server Function</i>
CN	<i>Core Network</i>
CP-OFDM	<i>Cycle Prefix-Orthogonal Frequency Division Multiplexing</i>
CT	<i>Core Network and Terminals</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DL	<i>Download</i>
DN	<i>Data Network</i>
DNN	<i>Data Network Name</i>
EDGE	<i>Enhanced Data rates for GSM Evolution</i>
eMBB	<i>Enhanced Mobile Broadband</i>
gNB	<i>gNodeB</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications</i>
GTP-U	<i>General Packet Radio Service Tunneling Protocol User</i>
HSPA+	<i>High Speed Packet Access Plus</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IIoT	<i>Industrial Internet of Things</i>
IKEv2	<i>Internet Key Exchange version 2</i>
IMSI	<i>International Mobile Subscriber Identity</i>
IMT-2000	<i>International Mobile Telecommunications-2000</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
IPsec	<i>IP Security Protocol</i>
ITU	<i>International Telecommunication Union</i>
LAN	<i>Local Area Network</i>
LI	<i>Legal Interception</i>
LTE	<i>Long Term Evolution</i>
LTS	<i>Long Term Support</i>
MAC	<i>Media Access Control</i>
ME	<i>Mobile Equipment</i>
MIMO	<i>Multiple Input Multiple Output</i>
mMIMO	<i>Massive Multiple Input Multiple Output</i>
mMTC	<i>Massive Machine-Type Communications</i>

MT	<i>Mobile Termination</i>
MTSO	<i>Mobile Telephone Switching Office</i>
MU-MIMO	<i>Multi User Multiple Input Multiple Output</i>
N3IWF	<i>Non-3GPP Interworking Function</i>
NaaS	<i>Network as a Service</i>
NAS	<i>Non-Access Stratum</i>
NAT	<i>Network Address Translation</i>
NF	<i>Network Function</i>
NFs	<i>Network Functions</i>
NFV	<i>Network Functions Virtualization</i>
NG-RAN	<i>Next Generation Radio Access Network</i>
NON3GPP	<i>NON Third Generation Partnership Project</i>
NR	<i>New Radio</i>
NRF	<i>Network Repository Function</i>
NSSAI	<i>Single-Network Slice Selection Assistance Information</i>
NSSF	<i>Network Slice Selection Function</i>
NTT	<i>Nippon Telegraph and Telephone</i>
NWDAF	<i>Network Data Analysis Function</i>
OFDM	<i>Orthogonal Frequency-Division Multiplexing</i>
OFDMA	<i>Orthogonal Frequency Division Multiplexing/Multiple Access)</i>
PCF	<i>Policy Control Function</i>
PDU	<i>Protocol Data Unit</i>
PFCP	<i>Packet Forwarding Control Protocol</i>
PLMN	<i>Public Land Mobile Network</i>
PSTN	<i>Public Switched Telephone Network</i>
PWS	<i>Public Warning System</i>
QoS	<i>Quality of Service</i>
RAN	<i>Radio Access Network</i>
SA	<i>System Architecture</i>
SC-FDMA	<i>Single-Carrier Frequency Division Multiple Access</i>
SCTP	<i>Stream Control Transmission Protocol</i>
SD	<i>Slice Differentiator</i>
SDN	<i>Software-Defined Networks</i>
SD-WAN	<i>Software-Defined Wide Area Network</i>
SIM	<i>Subscriber Identity Module</i>
SMF	<i>Session Management Function</i>
SNSSAI	<i>Single Network Slice Selection Assistance Information</i>
SSH	<i>Secure Shell</i>
SST	<i>Slice/Service Type</i>

SU-MIMO	<i>Single User Multiple-Input Multiple-Output</i>
TCP	<i>Transmission Control Protocol</i>
TE	<i>Terminal Equipment</i>
UDM	<i>User Data Management</i>
UDP	<i>User Datagram Protocol</i>
UDR	<i>Unified Data Repository</i>
UE	<i>User Equipament</i>
UICC	<i>Universal Integrated Circuit Card</i>
UL	<i>Upload</i>
UMTS	<i>Universal Mobile Telecommunications System</i>
UPF	<i>User Plane Function</i>
URLLC	<i>Ultra-reliable and Low-latency Communication</i>
USIM	<i>Universal Subscriber Identity Module</i>
VLAN	<i>Virtual LAN</i>
VM	<i>Virtual Machine</i>
VoIP	<i>Voice over IP</i>
WAN	<i>Wide Area Network</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
WLAN	<i>Wireless Local Area Network</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	PROBLEMATIZAÇÃO	15
1.2	OBJETIVOS GERAIS	16
1.3	OBJETIVOS ESPECÍFICOS	16
1.4	ESTRUTURA DO DOCUMENTO	17
2	REVISÃO DE LITERATURA	18
2.1	HISTÓRIA	18
2.2	ARQUITETURA DAS REDES MÓVEIS	19
2.3	3GPP	21
2.4	NON3GPP	22
2.4.1	N3IWF	23
2.4.2	NWu	23
2.5	QUINTA GERAÇÃO DE REDES MOVEIS	24
2.5.1	Núcleo do 5G	26
2.5.1.1	AMF	28
2.5.1.2	SMF	28
2.5.1.3	UPF	29
2.5.1.4	UDR	31
2.5.1.5	NRF	31
2.5.1.6	UDM	31
2.5.1.7	AUSF	32
2.5.1.8	PCF	32
2.5.1.9	NWDAF	32
2.5.1.10	NSSF	32
2.5.2	Tecnologias do 5G	33
2.5.2.1	Sessão PDU	34
2.5.2.2	Beamforming	35
2.5.2.3	MIMO	35
2.5.2.4	Fatiamento de Rede	36
2.5.3	Entendendo Contêineres, Docker e Kubernetes	38
2.5.4	VirtualBox	38
2.5.5	UERANSIM	39
2.5.6	Aether	39
2.5.7	SD-WAN	40
2.5.8	5G no meio industrial e IoT	41
3	SISTEMA PROPOSTO	42
3.1	PREPARAÇÃO DO AMBIENTE	43

3.2	INSTALAÇÃO DO AETHER	46
3.3	INSTALAÇÃO DO UERANSIM	48
3.4	CONFIGURAÇÃO DO N3IWF USANDO O PROJETO FREE5GC-COMPOSE	51
3.5	CONFIGURANDO O NWU	52
3.6	ALTERAÇÃO E ADAPTAÇÃO DO AMF E DO AUSF	52
3.7	FERRAMENTAS	53
3.8	METODOLOGIA DOS TESTES	54
4	RESULTADOS	56
4.1	DADOS OBTIDOS	59
4.1.1	UE-NON3GPP para Aether	59
4.1.2	UE-gNB para Aether	61
4.1.3	UE-gNB para UE-NON3GPP	62
4.1.4	UE-NON3GPP para UE-gNB	63
4.2	GRÁFICOS DOS DADOS OBTIDOS	64
5	CONCLUSÃO	68
	REFERÊNCIAS	69
	APÊNDICE A – Configuração do Aether	72
	APÊNDICE B – Configuração do UERANSIM	87

1 INTRODUÇÃO

A tecnologia 5G, mais recente geração de redes de comunicação móvel, visa proporcionar uma experiência de conectividade onipresente, garantindo maior velocidade, menor latência e capacidade para conectar um número muito maior de dispositivos simultaneamente. Para alcançar isso, as redes 5G precisam abranger uma ampla gama de tecnologias e padrões, incluindo aqueles fora do escopo do Projeto de Parceria em Terceira Geração 3GPP (*Third Generation Partnership Project*) que é o grupo responsável pela padronização das tecnologias e protocolos do 5G, como Wi-Fi e LTE. É nesse cenário que a integração da conectividade NON3GPP (*NON Third Generation Partnership Project*) nas redes 5G surge como um desafio importante.

Na era da Indústria 4.0, a Internet das Coisas Industrial (IIoT) está emergindo como uma força transformadora, com a capacidade de conectar uma enorme quantidade de dispositivos e sensores para melhorar a eficiência e a produtividade. Com a chegada do 5G, com sua capacidade excepcional de lidar com conexões massivas e fornecer alta velocidade e baixa latência, é um divisor de águas para a IIoT. No entanto, apesar do potencial transformador do 5G, existe um obstáculo para fazer todos os equipamentos com diferentes protocolos de comunicação se comunicarem e acessarem o núcleo do 5G.

Para abordar essa problemática, este trabalho se propõe a explorar e analisar a conexão NON3GPP em redes 5G, com foco na comparação entre as conexões 3GPP e NON3GPP. Serão utilizadas ferramentas como o UERANSIM (UERANSIM, 2023), um simulador de rede RAN aberto, o Aether (AETHER... , 2023), uma plataforma aberta do núcleo do 5G, o N3IWF, uma função de rede de interconexão sem fio NON3GPP, e o NWu, uma interface entre a rede de acesso NON3GPP e a rede 5G.

O UERANSIM será fundamental para a simulação de cenários de rede realistas 3GPP. Por meio do Aether, será possível automatizar e gerenciar eficientemente essas redes simuladas, permitindo uma avaliação mais precisa das características e do desempenho de cada tipo de conexão. A inclusão do N3IWF e NWu nesse ambiente permite uma análise detalhada das conexões NON3GPP, desde o estabelecimento e manutenção da conexão até a segurança e qualidade da comunicação.

Espera-se que a investigação a ser realizada neste trabalho forneça conhecimento valioso sobre o estado atual da integração de conexões NON3GPP em redes 5G e ajude a identificar oportunidades para futuras pesquisas e desenvolvimentos neste campo em rápido crescimento.

1.1 PROBLEMATIZAÇÃO

Embora a implementação do 5G na IIoT prometa uma revolução na comunicação entre dispositivos industriais, ela também apresenta desafios significativos. Um dos principais é a questão da compatibilidade entre dispositivos antigos e a nova tecnologia 5G.

A realidade econômica torna inviável a substituição completa dos dispositivos existentes por novos compatíveis com 5G. Portanto, é crucial encontrar uma solução que permita a coexistência e a comunicação eficaz entre dispositivos 5G e não 5G. A possibilidade de estabelecer uma conexão NON3GPP surge como uma solução promissora para este problema.

Um desafio é o gerenciamento da conectividade entre as redes variadas, englobando desde a seleção de rede, balanceamento de carga, até a continuidade da sessão e mobilidade entre redes. Devido às flutuações nas características do sinal, latência, capacidade e confiabilidade entre as redes 3GPP e NON3GPP, assegurar uma experiência de usuário consistente e de alta qualidade se torna uma tarefa complexa.

A segurança e a privacidade são questões cruciais. Surgem questões como a garantia da segurança nas conexões entre as redes 3GPP e NON3GPP ao integrar diferentes tecnologias, a eficácia no gerenciamento das credenciais do usuário entre as diversas redes, e a necessidade de assegurar a proteção contra ataques de interceptação ou *spoofing* em um ambiente de rede diversificado.

Além disso, não se pode ignorar questões práticas e de implementação. Considerações importantes incluem a melhor forma de os operadores de rede integrarem as redes 3GPP e NON3GPP, quais seriam os requisitos de *hardware*, *software* e infraestrutura, e como otimizar esses recursos para proporcionar a melhor experiência possível ao usuário.

Estes são apenas alguns dos desafios enfrentados ao conectar redes NON3GPP em redes 5G. A resolução desses problemas é essencial para o futuro das comunicações móveis e para a plena realização do potencial da rede 5G. Neste trabalho, será implementada e avaliada uma conexão NON3GPP com uma conexão 3GPP, visando entender melhor esses desafios e identificar possíveis soluções.

1.2 OBJETIVOS GERAIS

O objetivo geral deste trabalho é estabelecer e avaliar uma conexão entre o núcleo de rede 5G e dispositivos que não seguem o padrão adotado pelo 3GPP. Para alcançar esse objetivo, serão utilizados ambientes de simulação de redes de comunicação sem fio para testar a interoperabilidade, funcionalidade e desempenho dessa conexão. Espera-se que os resultados obtidos forneçam conhecimentos valiosos para uma compreensão mais aprofundada da integração de conexões NON3GPP em redes 5G, auxiliando no desenvolvimento e otimização de redes adaptadas a casos de uso que exigem a incorporação de dispositivos NON3GPP.

1.3 OBJETIVOS ESPECÍFICOS

- Estudar a arquitetura das redes de telecomunicação móvel.

- Analisar a rede móvel de quinta geração, seus principais componentes e serviços oferecidos.
- Configurar o ambiente de simulação Aether para acessar o núcleo do 5G;
- Utilizar o UERANSIM para emular dispositivos 3GPP e simular sua conexão com o núcleo 5G através do Aether.
- Utilizar o N3IWF e o NWU para emular dispositivos NON3GPP e simular sua conexão com o núcleo 5G através do Aether.
- Comparar a interoperabilidade e funcionalidade da conexão NON3GPP com a conexão 3GPP com o núcleo 5G.
- Realizar experimentos e análises de desempenho para avaliar a eficiência e qualidade da conexão NON3GPP.
- Analisar os resultados obtidos para contribuir com o conhecimento sobre a integração de conexões NON3GPP em redes 5G.
- Investigar a aplicabilidade dos resultados para o desenvolvimento e otimização de redes 5G em ambientes industriais e IoT.

1.4 ESTRUTURA DO DOCUMENTO

No Capítulo 2, são introduzidos conceitos básicos fundamentais para este trabalho, incluindo a história das redes móveis, a arquitetura das redes móveis e os protocolos de comunicação. Esse capítulo também detalha as diferenças e desafios associados às conexões 3GPP e NON3GPP. Além disso, a rede 5G, suas principais tecnologias e componentes constituintes são apresentados de maneira detalhada e por fim são apresentados detalhes sobre os *softwares* e ferramentas utilizadas para a simulação. O Capítulo 3 apresenta o sistema proposto e descreve os procedimentos e técnicas utilizadas para atingir os objetivos definidos, incluindo a configuração e simulação das conexões 3GPP e NON3GPP. No Capítulo 4, os resultados da simulação, incluindo a eficiência, confiabilidade e capacidade da conexão NON3GPP, são apresentados e discutidos. Por fim, o Capítulo 5 oferece uma avaliação geral dos resultados obtidos, discute as implicações desses resultados para o campo das redes de telecomunicações móveis e meio industrial.

2 REVISÃO DE LITERATURA

Neste capítulo, são abordados os conceitos cruciais das tecnologias empregadas ao longo deste trabalho. Começamos com a visão histórica da evolução de suas gerações e das tecnologias-chave aplicadas em cada uma delas e uma explicação dos princípios fundamentais da arquitetura das redes de telecomunicações móveis. Posteriormente, detalhamos a quinta geração da rede móvel, discutindo suas principais tecnologias e a relevância delas para a pesquisa. Concluimos com a apresentação dos *softwares* e ferramentas empregadas na realização deste trabalho, além de discutir algumas de suas aplicações.

2.1 HISTÓRIA

Este capítulo traça a evolução histórica das redes móveis, desde as suas origens até a modernidade, e destaca as inovações chave que cada geração trouxe.

A jornada começa com a primeira geração 1G, implantada inicialmente no Japão em 1979 pela NTT (*Nippon Telegraph and Telephone*) e posteriormente disponibilizada comercialmente em outros países na década de 1980. Essa geração, baseada em sistemas analógicos, foi projetada para fornecer serviços de comunicação de voz. Uma inovação significativa foi o MTSO (*Mobile Telephone Switching Office*), que facilitava o *roaming* (a capacidade de um dispositivo manter a conectividade ao se mover entre áreas geográficas) e o *handoff* (o processo de transferência de uma chamada em andamento ou sessão de dados de uma célula de serviço para outra sem interrupção), tornando-se a espinha dorsal da infraestrutura 1G (BOTH, C. ET AL, 2020).

A década seguinte viu o nascimento da segunda geração 2G, marcada por uma mudança revolucionária para a transmissão de dados digitais. A introdução da internet móvel e das mensagens de texto, juntamente com o aprimoramento da qualidade do serviço e eficiência espectral, destacaram a 2G. Além disso, surgiram evoluções como o GPRS (*General Packet Radio Service*) e o EDGE (*Enhanced Data rates for GSM Evolution*), permitindo maior capacidade de transmissão de dados e estabelecendo a base para as gerações futuras (BOTH, C. ET AL, 2020).

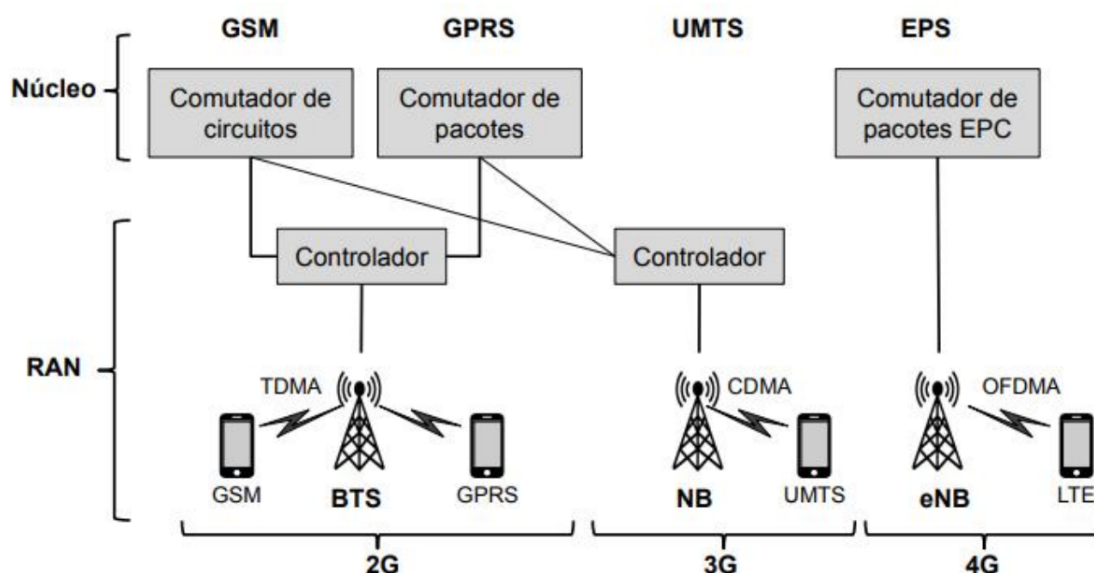
Respondendo à crescente demanda por altas taxas de transmissão para serviços multimídia, a terceira geração 3G foi desenvolvida no final dos anos 1990, seguindo o conjunto de requisitos IMT-2000 (*International Mobile Telecommunications-2000*) definido pela ITU (*International Telecommunication Union*). Entre as especificações apresentadas, o padrão UMTS (*Universal Mobile Telecommunications System*) da 3GPP (*Third Generation Partnership Project*) destacou-se, levando a uma evolução natural do GSM e a introdução do HSPA+ (*High Speed Packet Access Plus*), que aprimorou dramaticamente as taxas de transferência de dados em relação às suas predecessoras (DAVANÇO SILVA, G. H., 2021).

Finalmente, a quarta geração 4G foi lançada em 2010, elevando a capacidade de

transmissão de dados a um novo patamar e introduzindo o conceito de Voz sobre IP (VoIP (*Voice over IP*)). Esta geração foi padronizada através do LTE, da 3GPP, e do WiMAX (*Worldwide Interoperability for Microwave Access*), do IEEE, ambos baseados em multiplexação ortogonal por divisão de frequência (OFDMA (*Orthogonal Frequency Division Multiplexing/Multiple Access*)) e SC-FDMA (*Single-Carrier Frequency Division Multiple Access*) (KRAUS, D, 2021).

Ao longo de cada etapa dessa evolução, novas tecnologias e atualizações foram integradas, muitas vezes tornando os limites entre as gerações um pouco nebulosos. No entanto, cada geração introduziu avanços significativos em termos de eficiência, capacidade e qualidade do serviço, pavimentando o caminho para uma variedade cada vez maior de aplicações móveis. As principais mudanças podem ser vista na Figura 1.

Figura 1 – Evolução arquitetural das redes móveis.



Fonte: (BOTH, C. ET AL, 2020)

2.2 ARQUITETURA DAS REDES MÓVEIS

A arquitetura de redes móveis constitui um conjunto sofisticado de protocolos e tecnologias, habilitando dispositivos móveis como smartphones e tablets a se conectarem a redes de comunicação sem fio. Isso abrange desde redes de celular até serviços de dados, promovendo a troca de informações. A estrutura dessa arquitetura é estratificada, com cada camada desempenhando funções específicas.

A administração dessa infraestrutura de telecomunicações móveis é realizada por uma operadora de rede, formalmente conhecida como PLMN (*Public Land Mobile Network*). A Figura 2 ilustra seus quatro componentes essenciais: a CN (*Core Network*), a RAN (*Ra-*

Radio Access Network), o sistema de gerenciamento e o equipamento de usuário, comumente chamado de móvel e formalmente referido como UE (COX, 2020).

No contexto das redes 5G, o termo UE é utilizado de forma geral para descrever qualquer sistema que possua conectividade com uma rede 5G. Isso inclui tanto equipamentos estáticos, como servidores em *data centers* ou sensores em aplicações industriais, quanto sistemas embarcados em dispositivos móveis, como smartphones e drones. Além disso, aplicações virtualizadas, como máquinas virtuais, *containers* e servidores virtuais, também são considerados equipamentos de usuário na rede 5G.

Essa terminologia é importante para entendermos que, com a implementação da rede 5G, a conectividade não se restringe apenas a dispositivos móveis, mas também se estende a diversos tipos de equipamentos e aplicações, possibilitando novas oportunidades e possibilidades de uso para a tecnologia.

O UE é composto por dois componentes. O primeiro é o equipamento responsável pela comunicação, conhecido como ME (*Mobile Equipment*). Em algumas circunstâncias, este componente pode ser subdividido em dois dispositivos distintos: o MT (*Mobile Termination*), responsável por todas as funções de comunicação, e o TE (*Terminal Equipment*), que é o equipamento em si. O segundo componente é o UICC (*Universal Integrated Circuit Card*), também referido como SIM (*Subscriber Identity Module*) Card. Esse componente aloja o USIM (*Universal Subscriber Identity Module*), que armazena dados específicos do usuário (MAZIERO BARETTA, J. L., 2022).

A CN (*Core Network*) desempenha a função crucial de transportar o tráfego entre o dispositivo móvel e uma ou mais redes externas, tais como a Rede Telefônica Pública Comutada (PSTN) ou a *Internet*. A CN também supervisiona as comunicações do dispositivo móvel com essas redes externas e armazena informações sobre os assinantes da operadora de rede.

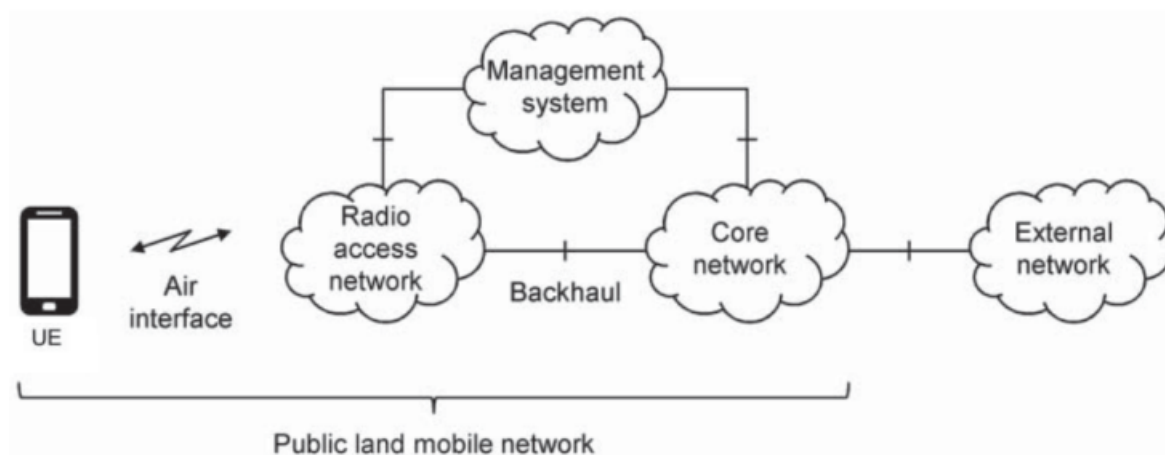
Por sua vez, a RAN gerencia as comunicações de rádio entre a rede e o dispositivo móvel. A RAN estabelece comunicação com a CN através de uma conexão conhecida como *backhaul*, e com o dispositivo móvel através de uma interface de rádio. Nesta interface, a direção da rede para o dispositivo móvel é conhecida como *downlink*, e a direção do dispositivo móvel para a rede é o *uplink* (COX, 2020). A RAN consiste em estações-base, que transmitem e recebem informações em uma ou mais frequências de rádio, definidas pela operadora. Cada frequência está associada a uma largura de banda, que representa a extensão do sinal de rádio naquele espectro. Por exemplo, uma estação base 5G operando a uma frequência de 3,5 GHz e com uma largura de banda de 40 MHz, transmite na faixa de 3480 a 3520 MHz. As operadoras de telefonia móvel têm permissão para transmitir em uma faixa definida do espectro licenciado, após a obtenção de uma licença para isso (COX, 2020).

Um sistema de gerenciamento separado comanda a rede e suas tarefas incluem configurar os vários componentes da rede, monitorar seu desempenho, relatar eventuais

falhas à operadora de rede e faturar o usuário.

Um dispositivo móvel tem a capacidade de se comunicar fora da área de cobertura da operadora de rede utilizando os recursos de dois PLMNs: a rede visitada, onde o dispositivo móvel se encontra, e a rede doméstica da operadora. Esse tipo de situação é conhecido como roaming (COX, 2020).

Figura 2 – Arquitetura de um sistema de telecomunicações móveis.



Fonte: (COX, 2020).

2.3 3GPP

O 3GPP (*Third Generation Partnership Project*) é uma organização que desenvolve e mantém padrões técnicos para redes móveis e sistemas de comunicação. A estrutura organizacional do 3GPP é dividida em grupos de trabalho que se concentram em diferentes aspectos das tecnologias de rede móvel. Será detalhado alguns dos principais grupos de trabalho e suas respectivas funções:

- **RAN (*Radio Access Network*)**: este grupo é responsável pela padronização da interface aérea entre os dispositivos móveis e as estações base;
- **SA (*System Architecture*)**: este grupo trabalha na arquitetura geral do sistema, serviços e recursos das redes móveis;
- **CT (*Core Network and Terminals*)**: este grupo é responsável pela padronização das funcionalidades de rede central e dos terminais móveis.

Esses grupos de trabalho colaboram entre si para desenvolver as especificações técnicas e os relatórios necessários para garantir a interoperabilidade entre os sistemas de telecomunicações, promovendo a evolução das redes móveis e a introdução de novos serviços e recursos.

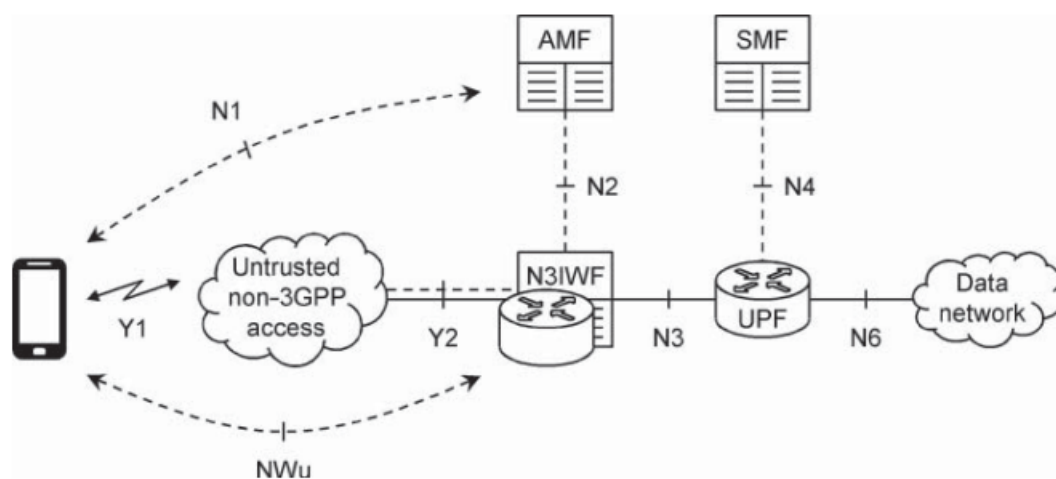
O 3GPP também está envolvido no desenvolvimento de várias gerações de padrões de rede móvel, como no 3GUMTS (*Universal Mobile Telecommunications System*): O UMTS foi o primeiro padrão desenvolvido pelo 3GPP, 4G (LTE): O LTE e 5G NR (*New Radio*): O NR é a mais recente geração de padrões de rede móvel (PENTTINEN, 2019).

2.4 NON3GPP

O termo NON3GPP se refere a redes de comunicação que não são padronizadas pelo 3GPP. Isso inclui uma ampla variedade de tecnologias e redes sem fio que não seguem os padrões e especificações estabelecidos, como redes de acesso sem fio não licenciadas como *Wi-Fi*, redes *ad-hoc*, redes satelitais, redes de sensores sem fio, entre outras.

Os UE são capazes de se conectar à rede principal 5G por meio de redes de acesso NON3GPP, como o IEEE Wi-Fi 802.11, usado para redes locais sem fio WLANs. A versão 15 realizada pela 3GPP oferece suporte para acesso em redes sem fio não confiáveis, em que a rede principal 5G assume que a rede de acesso não é segura. Já a versão 16 estende esse suporte para redes sem fio confiáveis e para acesso com fio. Embora o acesso NON3GPP começou a ser especificado na versão 8 do padrão 3GPP durante a era do 4G, a arquitetura 5G está mais alinhada com o acesso. A Figura 3 apresenta as funções de rede mais importantes nessa arquitetura (COX, 2020)

Figura 3 – Arquitetura de acesso para rede NON3GPP.



Fonte: (COX, 2020).

Nessa arquitetura, o termo 5G-AN (*5G Access Network*) abrange tanto as redes de acesso 3GPP NG-RAN quanto as redes de acesso NON3GPP, como Wi-Fi. Em ambos os casos, o servidor AMF (*Access and Mobility Management Function*) controla o dispositivo móvel e realiza a autenticação do mesmo junto ao AUSF (*Authentication Server Function*). O AMF se comunica com o dispositivo móvel através do ponto de referência N1 e com a

rede de acesso através do ponto de referência N2, utilizando os mesmos procedimentos de sinalização que utiliza para o acesso 3GPP (COX, 2020).

2.4.1 N3IWF

No núcleo do 5G existe uma função de rede nova, conhecida como N3IWF (*Non-3GPP Interworking Function*), que possibilita a incorporação de redes que não estão incluídas no acesso estabelecido pelo 3GPP, isto é, permite a conexão das redes NON3GPP ao núcleo da rede 5G.

O N3IWF cria novos pontos de referência N2 e N3. Essa função atua como uma interface entre a rede de acesso e o núcleo 5G, protegendo as comunicações do dispositivo móvel na rede de acesso. Para tanto, estabelece um túnel seguro para o dispositivo móvel utilizando mensagens de sinalização sobre um ponto de referência chamado *NWu* (COX, 2020).

Essa função é geralmente implementada em *gateways* ou pontos de interconexão. Essa interconexão é feita através do estabelecimento de túneis IKEv2 (*Internet Key Exchange version 2*) e IPsec (*IP Security Protocol*) entre N3IWF e UE (MAILER, C, 2020). A Figura 3 ilustra a utilização do N3IWF para conectar uma rede não 3GPP ao núcleo 5G.

2.4.2 NWu

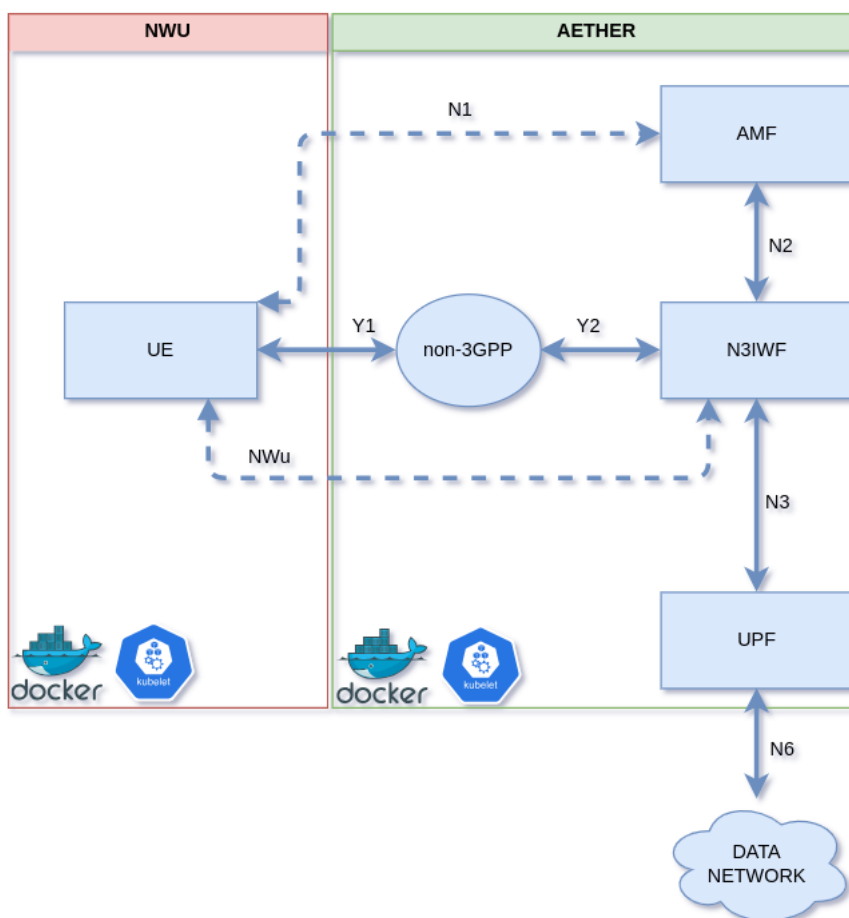
NWu é um componente chave na configuração de uma rede móvel que conecta dispositivos através de acessos NON3GPP, a uma rede móvel 3GPP, como uma rede 5G.

O *NWu* é um túnel seguro e criptografado IPsec (*IP Security Protocol*) que é estabelecido entre o UE e um elemento de rede chamado N3IWF (*Non-3GPP Interworking Function*). O propósito deste túnel *NWu* é permitir que o tráfego de sinalização e de dados seja encaminhado entre o dispositivo e a rede móvel.

O *NWu* é estabelecido sobre duas outras conexões, Y1 e Y2, que o UE utiliza para se conectar à rede NON3GPP e depois à *Internet*, respectivamente. A conexão Y1 é tipicamente uma interface aérea *WiFi*, e a Y2 pode ser a própria *Internet* pública. Essas conexões Y1 e Y2 não são controladas nem especificadas pela 3GPP.

Em outras palavras, o *NWu* é uma solução de tunelamento seguro que permite a um dispositivo usar acessos de redes NON3GPP para se conectar a uma rede móvel 3GPP. O tráfego de dados é roteado através deste túnel *NWu*, de modo que o dispositivo possa interagir com a rede móvel como se estivesse conectado diretamente a ela.

Figura 4 – Sistema Nwu-Aether.



Fonte: O autor.

2.5 QUINTA GERAÇÃO DE REDES MOVEIS

O 5G é a quinta geração de redes de telecomunicações móveis, que promete revolucionar a maneira como nos conectamos e interagimos com a tecnologia. Com velocidades de dados extremamente rápidas, latência ultrabaixa e capacidade de conexão de milhões de dispositivos simultaneamente, o 5G tem o potencial de impulsionar uma ampla gama de aplicações, desde veículos autônomos e cidades inteligentes até saúde digital e realidade virtual. Além disso, o 5G oferece uma nova gama de categorias de serviços, projetadas para atender às necessidades específicas de diferentes setores, como indústria, agricultura, transporte e entretenimento. No entanto, o 5G também traz desafios técnicos, regulatórios e de segurança, que precisam ser abordados para aproveitar todo o seu potencial. O 5G representa uma evolução significativa nas redes de comunicação móvel, com o potencial de transformar a sociedade e a economia em escala global.

As redes 5G são uma evolução das redes LTE, oferecendo taxas de transmissão de dados mais altas e menor latência de ponta a ponta. Essas características são altamente

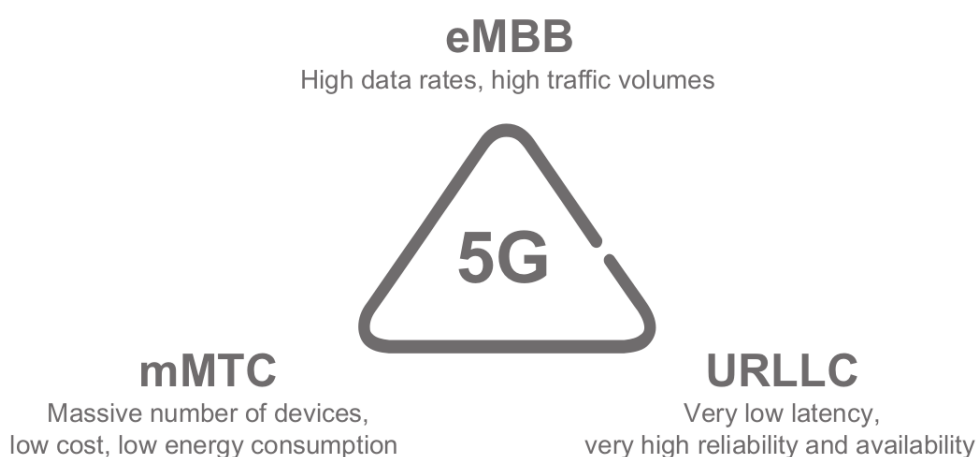
desejadas devido ao aumento significativo de dispositivos multimídia conectados, e a tendência de aumento dessa demanda nos próximos anos.

No entanto, o 5G se diferencia das gerações anteriores em outros aspectos. Uma mudança importante é o novo paradigma em que as operadoras, que antes tinham os usuários finais como principais clientes, podem passar a ter as indústrias como seus principais clientes. Isso representa uma mudança sem precedentes, não apenas em termos tecnológicos, mas também no modelo de negócios (ROMMER, 2020).

Outra mudança significativa entre o 4G e o 5G é a criação de novas categorias de serviços que visam atender a diversos cenários, aplicações e casos de uso. As três principais categorias podem ser vista na Figura 5.

- **eMBB (*Enhanced Mobile Broadband*)**: cujo objetivo é fornecer altas velocidades de *downlink*, de até 1 Gbit/s, para áreas metropolitanas densamente povoadas;
- **mMTC (*Massive Machine-Type Communications*)**: que possibilita a comunicação máquina-máquina e aplicações de Internet das Coisas (IoT), atendendo às demandas de uma nova geração de dispositivos sem fio, sem comprometer outras classes de serviço;
- **URLLC (*Ultra-reliable and Low-latency Communication*)**: que permite a comunicação crítica com requisitos de latência extremamente baixa, de até 1 ms ou menos, sendo especialmente adequada para aplicações sensíveis ao tempo, onde a largura de banda não é tão importante.

Figura 5 – Categorias de serviço.



Fonte: (ROMMER, 2020).

É importante ressaltar que a categorização dos casos de uso do 5G em três classes distintas é uma simplificação utilizada para definir os requisitos da tecnologia. Nem todos os casos de uso se encaixam perfeitamente em uma dessas classes. Por exemplo, pode haver

serviços que demandem alta confiabilidade, mas que não tenham requisitos de latência tão críticos. Da mesma forma, pode haver casos de uso que necessitem de dispositivos de baixo custo, mas onde a duração da bateria não seja tão relevante. A classificação em classes de serviço é apenas uma maneira de organizar as diferentes necessidades e requisitos dos casos de uso do 5G, mas, na prática, muitos cenários podem apresentar características que não se enquadram exclusivamente em uma única classe (ROMMER, 2020).

Além disso, o 5G oferece oportunidades para as operadoras fornecerem conectividade para residências em áreas onde a implantação de fibra óptica é difícil ou custosa. Isso evita a necessidade de obras demoradas e dispendiosas para instalação desse tipo de acesso.

Embora o aumento da taxa de dados seja um dos principais objetivos do 5G em relação às gerações anteriores, um aspecto igualmente importante é a capacidade de gerenciar inúmeros dispositivos comunicando-se simultaneamente (PENTTINEN, 2019).

Tabela 1 – Requisitos de serviço para o 5G.

Descrição	Valor
Velocidades de dados máximas	Até 20 Gbit/s de DL, até 10 Gbit/s de UL
Média de velocidades de dados	Até 100 Mbit/s DL, Até 50 Mbit/s UL
Eficiência espectral	Até 30 bits/s/Hz DL, Até 15 bits/s/Hz glsUL
Densidade de conexão	Até 1 milhão devices/km ²
Vida útil da bateria	Mais de 10 anos
Mobilidade	Até 500 km/h
Latência do usuário	1ms para uso industrial, 4 ms para Banda Larga Móvel
Confiabilidade	Pelo menos 99.999%

Fonte: (ROMMER, 2020)

2.5.1 Núcleo do 5G

O núcleo das redes 5G precisa conseguir suportar o novo paradigma da arquitetura baseada em serviços para redes modulares e proporcionar uma experiência consistente aos usuários de redes de acesso 3GPP ou NON3GPP. Além disso, deve ser adaptável a tecnologias baseadas em nuvem, computação de borda e serviços de comunicação máquina-máquina, a fim de viabilizar a utilização de carros autônomos e linhas de montagem industrial robotizadas entre outras aplicações.

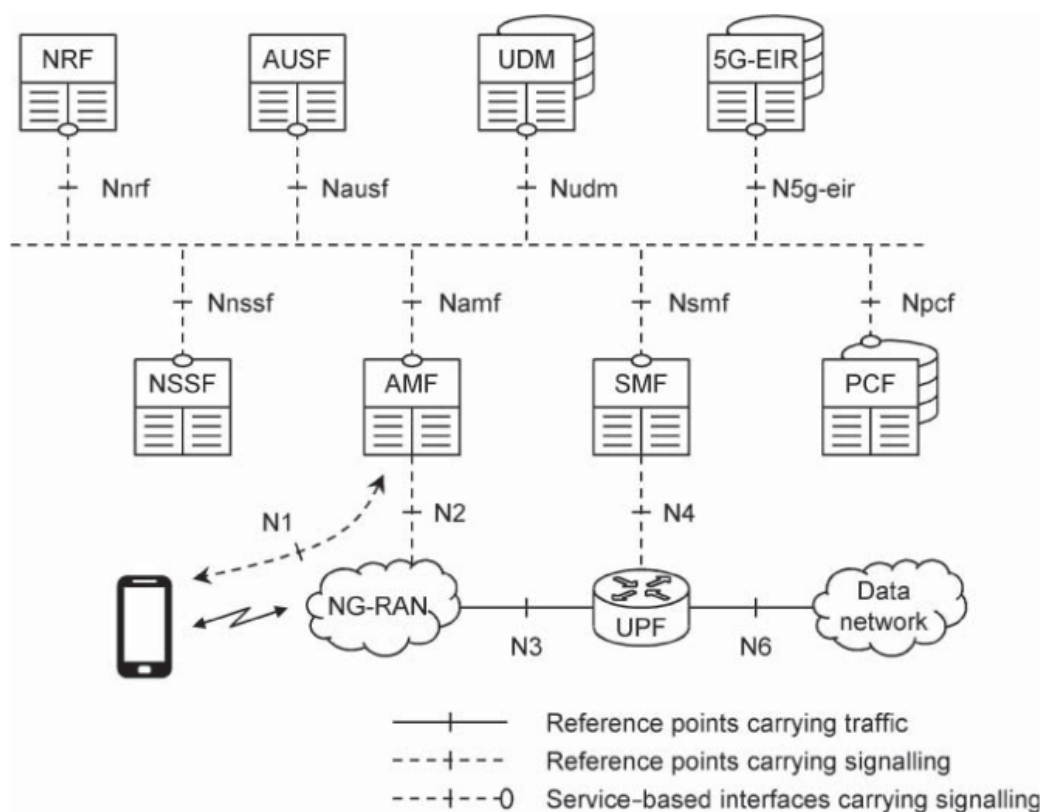
Uma das novidades introduzidas pelo 5G é o conceito de interfaces baseadas em serviços, o que significa que as funções de rede, que incluem lógica e funcionalidade para processar fluxos de sinalização, não são interconectadas por meio de interfaces ponto-a-ponto, mas sim expõem e tornam outros serviços disponíveis para outras funções de rede. Em cada interação entre as funções de rede, uma atua como consumidora de serviço e a outra como produtora do serviço. Essa abordagem baseada em serviços é uma grande

diferença em relação às gerações anteriores e suas arquiteturas tradicionais (ROMMER, 2020).

As funções de rede consumidoras podem localizar e contatar uma função produtora de serviço por meio do conceito de descoberta de serviço. Dentro da arquitetura 5G, esse registro de consumidoras e produtoras é feito por uma função de rede dedicada chamada Função do Repositório de Rede (NRF). Outra forma pela qual as funções de rede podem interagir é por meio do conceito de inscrição e notificação de serviços, possibilitando que uma função produtora de serviço envie notificações para todos os consumidores quando um critério pré-estabelecido é alcançado. Essa estratégia evita que os consumidores enviem mensagens periódicas para solicitar o serviço, permitindo que o produtor apenas os notifique quando necessário (DAVANÇO SILVA, G. H., 2021).

Embora existam diversas Funções de Rede (NFs) no núcleo 5G, aquelas utilizadas durante o processo de simulação estão descritas a seguir e podem ser visualizadas na Figura 6.

Figura 6 – Funções de rede essenciais do núcleo 5G.



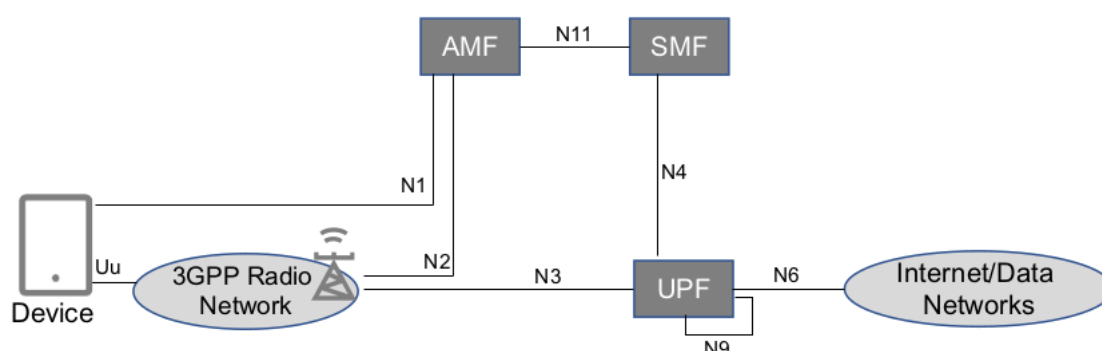
Fonte: (COX, 2020).

2.5.1.1 AMF

O AMF (*Access and Mobility Management Function*) é responsável por estabelecer conexões com a RAN e o UE, através das interfaces N2 e N1, respectivamente (ROMMER, 2020). O AMF executa a gestão do cadastro, autenticação e movimentação do UE, assegurando ainda a criptografia e a integridade das mensagens NAS. Além disso, o AMF retransmite mensagens de gerenciamento de sessão, como, por exemplo, entre o UE e o SMF.

O AMF também suporta a entrega de mensagens de aviso público (PWS) e possui uma interface para interceptação legal (LI) (MAILER, C, 2020). É importante destacar que um determinado UE só pode ser atendido por um AMF em um determinado momento. A Figura 7 ilustra as interfaces utilizadas pelo AMF para interagir com as demais funções da rede.

Figura 7 – Interfaces de rede utilizadas pelo AMF.



Fonte: (ROMMER, 2020).

2.5.1.2 SMF

A SMF (*Session Management Function*) é responsável por estabelecer e gerenciar a conexão entre o UE e a rede de dados (DN), selecionando a UPF (*User Plane Function*) conforme os requisitos da conexão e estabelecendo uma sessão PDU entre o acesso e o UPF, ou entre UPFs quando necessário (CHANDRAMOULI; LIEBHART; PIRSKANEN, 2019).

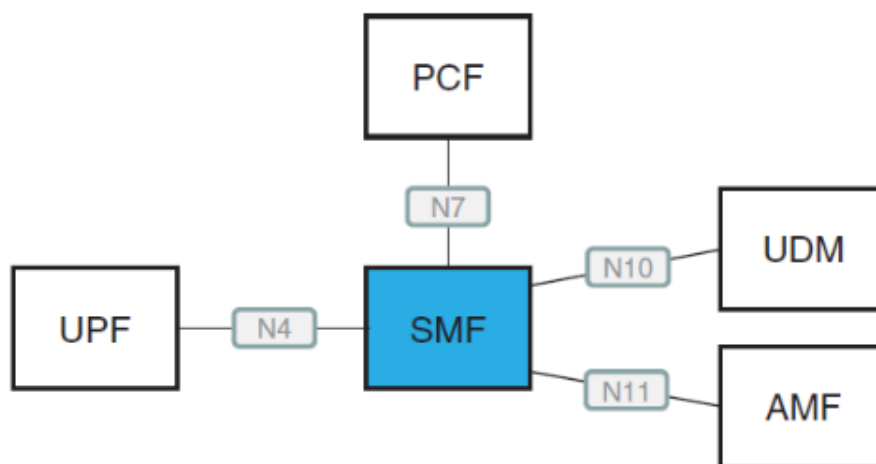
A interface de comunicação entre o SMF e o UPF (*User Plane Function*) é chamada de N4. O protocolo PFCP, desenvolvido para o 4G e evoluído para o 5G, é utilizado na camada de aplicação da interface N4 e atua em cima do protocolo UDP (PENTTINEN, 2019).

O estabelecimento da sessão PDU, assim como suas características, depende dos requisitos do UE, das informações dos bancos de dados do UDM/UDR e das políticas de serviço e QoS configuradas na PCF (ROMMER, 2020). Além disso, o SMF pode alocar

endereços IP para as sessões PDU e possui suporte para a interface de interceptação legal (ROMMER, 2020).

A Figura 8 demonstra as interfaces empregadas pelo SMF para interação com outras funções da rede, sublinhando sua relevância no gerenciamento das sessões de dados entre o UE e a DN no cenário do 5G.

Figura 8 – Interfaces de rede utilizadas pelo SMF.



Fonte: (PENTTINEN, 2019).

2.5.1.3 UPF

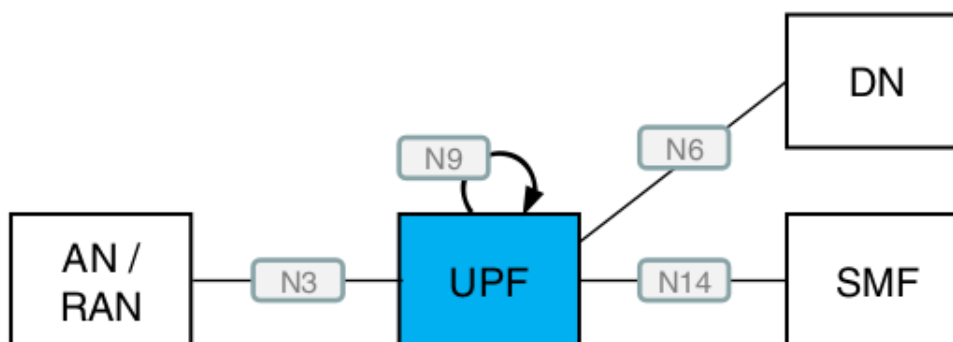
A função de Plano de Usuário (UPF), controlada pela SMF, é responsável pelo processamento e encaminhamento de dados de usuário em uma rede de telecomunicações. A UPF é conectada a redes *IP* externas e atua como um ponto de ancoragem estável para os equipamentos em relação a essas redes externas. Além disso, a UPF desempenha várias funções, tais como a geração de relatórios de uso de dados para a SMF, que são posteriormente incluídos nos relatórios de tarifação e outros fins (ROMMER, 2020).

As instâncias UPF oferecem não apenas a função de interconexão entre a sessão PDU e as redes de dados, mas também outras funções, tais como:

- Detecção de tráfego, incluindo detecção de aplicações: A UPF é capaz de identificar o tipo de tráfego que está sendo transmitido, incluindo a detecção de aplicações específicas, permitindo a aplicação de políticas de gerenciamento de tráfego adequadas;
- Encaminhamento de tráfego baseado em regras definidas pela SMF: A UPF recebe regras da SMF (*Session Management Function*) e encaminha o tráfego de acordo com essas regras, garantindo que os dados do usuário sejam direcionados corretamente na rede;

- Carregamento de dados e encaminhamento para garantir a integridade dos dados durante um *handover*: Durante um *handover* de uma conexão de rede para outra, a UPF é responsável por garantir que os dados do usuário sejam transferidos de forma segura e sem perda de integridade;
- Aplicação de *QoS*: A UPF é capaz de aplicar políticas de Qualidade de Serviço (QoS) para garantir que os recursos de rede sejam alocados de forma adequada, priorizando determinados tipos de tráfego conforme as políticas definidas;
- Relatório de uso de recursos: A UPF gera relatórios de uso de recursos para a SMF, por meio da interface N4, permitindo o monitoramento e a análise do consumo de recursos de rede pelos usuários;
- Relatório de eventos baseados em gatilhos e condições: A UPF pode gerar relatórios de eventos com base em gatilhos e condições específicas, permitindo a detecção de eventos anormais ou importantes na rede (DAVANÇO SILVA, G. H., 2021);
- Replicação do tráfego do plano de usuário para monitoramento: A UPF pode replicar o tráfego do plano de usuário para fins de monitoramento, permitindo a análise do tráfego em tempo real para fins de diagnóstico e solução de problemas.

Figura 9 – Interfaces de rede utilizadas pelo UPF.



Fonte: (PENTTINEN, 2019).

Além disso, a UPF também é responsável pela aplicação de QoS e marcação de pacotes em direção à rede de acesso via rádio ou em direção às redes de dados, permitindo que a rede de transporte lide com cada pacote de forma individual e com a devida prioridade em caso de congestionamento na rede.

Em termos de arquitetura, o plano de usuário consiste na conexão inicial da UE com o plano de usuário, por meio da tecnologia de acesso utilizada (por exemplo, NG-RAN). Do ponto de vista da rede de acesso, o plano de usuário é a conexão entre o núcleo

da rede, por meio das interfaces N3 ou N9, até a rede de dados, via N6, como pode ser observado na Figura 9.

2.5.1.4 UDR

A UDR (*Unified Data Repository*) é um banco de dados onde diferentes tipos de dados são armazenados de forma centralizada. Esses dados podem incluir informações de assinante, como perfis de assinante e políticas de rede ou de usuário. A UDR é projetada para ser usada como um serviço por outras funções de rede, como a UDM e a PCF, permitindo o armazenamento e acesso eficiente de dados em um único local. A UDR desempenha um papel importante na gestão e controle de políticas de rede e dados de assinantes em arquiteturas de redes de telecomunicações, proporcionando uma abordagem unificada e eficaz para o armazenamento e acesso a dados relevantes (ROMMER, 2020).

2.5.1.5 NRF

O NRF (*Network Repository Function*) é um repositório onde as funções de rede (NFs) disponíveis para o núcleo são armazenadas. Ele registra as características de cada NF e permite que outras NFs consultem seu banco de dados para obter o endereço de rede dos serviços desejados (ROMMER, 2020). Um benefício significativo do NRF é que ele dispensa a necessidade de conhecimento antecipado dos endereços e perfis dos elementos do núcleo, possibilitando que sejam requisitados quando uma determinada NF precisa de um serviço proveniente de outra NF. Cada NF é responsável por se registrar no NRF e atualizar seu *status*, e o NRF possui um mecanismo de "*keep alive*" que identifica elementos que não estão mais disponíveis (MAILER, C, 2020).

Alterações na estrutura do núcleo ou na capacidade de escalabilidade podem ser realizadas unicamente através de atualizações do *status* da NF modificada no NRF, eliminando a necessidade de mudar as configurações internas das outras NFs. Os dados disponíveis no NRF abrangem o tipo de NF, ID, endereço, capacidade, serviços suportados e informações de autorização (PENTTINEN, 2019).

2.5.1.6 UDM

A UDM (*User Data Management*) é responsável por gerar dados para autenticar os equipamentos da rede e autorizar o acesso de usuários específicos com base em dados de inscrição. Isso pode significar aplicar diferentes regras de acesso para usuários remotos e locais. Além disso, quando há mais de uma instância de AMF e SMF (Session Management Function) na rede, a UDM mantém um registro de qual instância está servindo a qual equipamento em específico (ROMMER, 2020).

2.5.1.7 AUSF

O AUSF (*Authentication Server Function*) tem a responsabilidade de conduzir a autenticação primária e a criação de chaves entre o UE e a rede. Este processo se dá através do uso de informações obtidas do UDM. O AUSF desempenha um papel importante na segurança da rede, garantindo que apenas dispositivos autorizados possam acessar a rede e estabelecer conexões seguras (PENTTINEN, 2019).

2.5.1.8 PCF

O núcleo 5G é composto por diversas funções de rede, e uma das peças centrais é a PCF (*Policy Control Function*), que atua como interface com outras NFs. Essa função é responsável por gerar políticas de controle para o gerenciamento de sessões de acesso e mobilidade, seleção de acesso do usuário (UE) e seleção de sessões PDU, bem como suporte para transferências em background (COX, 2020).

As funcionalidades da PCF são vitais para o controle de políticas relacionadas ao usuário, seus serviços e sessões de dados. Além disso, essas funcionalidades permitem definir quais tecnologias de acesso de rádio um usuário pode utilizar, oferecendo maior flexibilidade e controle para as operadoras de telecomunicações.

A PCF é uma peça fundamental para garantir a eficiência e qualidade da rede 5G, permitindo que os usuários possam acessar serviços e aplicativos de forma rápida e segura.

2.5.1.9 NWDAF

A função NWDAF (*Network Data Analysis Function*) é responsável por coletar dados de outras funções de rede por meio de serviços que expõem eventos dessas funções. Ela também coleta dados de sistemas, operações e gerenciamento e do repositório unificado de dados (UDR). Qualquer outra função de rede ou até mesmo aplicativos externos podem teoricamente consumir os serviços oferecidos pela NWDAF (DAVANÇO SILVA, G. H., 2021).

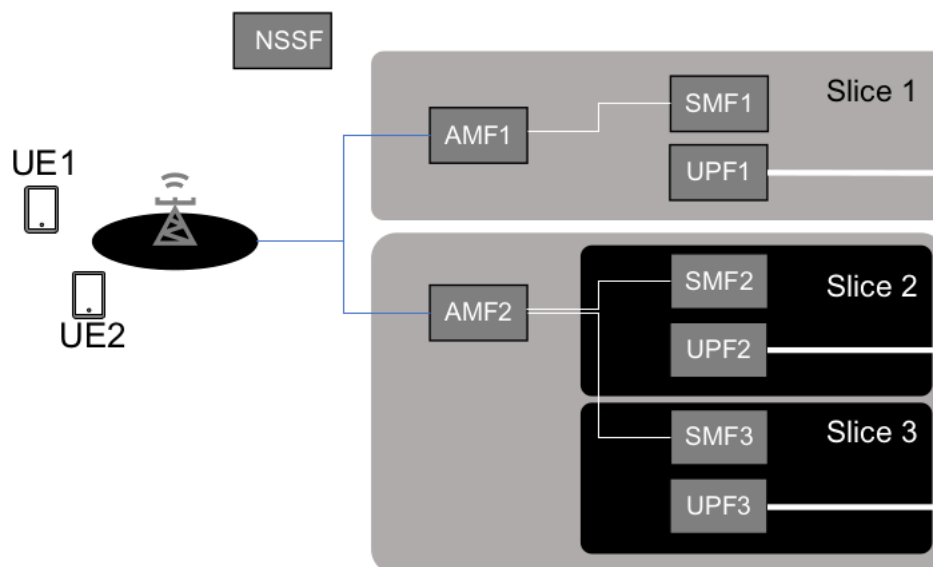
Os principais consumidores da NWDAF são a NSSF e a PCF. Ao coletar dados, a NWDAF pode realizar análises, como resumos históricos ou estatísticos, ou previsões de valores futuros. As análises realizadas pela NWDAF podem ser usadas por outras funções de rede para realizar ações específicas na rede, como modificar uma fatia específica ou modificar a qualidade de serviço (QoS) de um serviço (ROMMER, 2020).

2.5.1.10 NSSF

A NSSF (*Network Slice Selection Function*) é responsável pela seleção de fatias de rede com base em valores combinados de informações de assistência de seleção de fatia de rede única (SNSSAI). Seu único propósito é permitir que as fatias desejadas sejam acessadas através da assinatura (autorização ou permissão concedida ao usuário

para acessar as fatias de rede selecionadas). Na Figura 10, é apresentado um exemplo simplificado de fatiamento de rede. O dispositivo UE1 se conecta à fatia 1, composta por AMF, SMF e UPF dedicadas, enquanto o dispositivo UE2 se conecta simultaneamente às fatias 2 e 3, cada uma contendo uma SMF e uma UPF, mas sendo servidas por uma AMF2 em comum. (ROMMER, 2020).

Figura 10 – Interfaces de rede utilizadas pelo NSSF.



Fonte: (ROMMER, 2020).

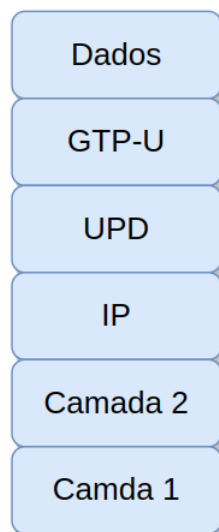
2.5.2 Tecnologias do 5G

A 3GPP define uma entidade lógica chamada gNB. Essa designação é utilizada somente quando se refere a uma estação rádio-base NR conectada a um núcleo 5G. As estações base são interconectadas pela interface Xn, que consiste na parte de sinalização Xn-C e na transferência de dados Xn-U. Todas as estações base estão conectadas a uma ou mais AMFs e UPFs no núcleo da rede pelas interfaces N2 e N3 (ROMMER, 2020).

Os dados de usuários são transferidos entre estações base e UPFs utilizando redes IP. A pilha de protocolos completa é mostrada na Figura 11 e é a mesma para as interfaces Xn-U e N3 (DAVANÇO SILVA, G. H., 2021). O GTP-U (*General Packet Radio Service Tunneling Protocol User*) é um protocolo bem conhecido, uma vez que é utilizado em todas as gerações de sistemas móveis anteriores e proporciona um serviço de comunicação de dados confiável. O GTP-U é utilizado por cima de uma pilha UDP/IP convencional que, por sua vez, é executada por cima de protocolos de camada 2, tipicamente *Ethernet*.

A 3GPP escolheu a forma de onda CP-OFDM (*Cycle Prefix-Orthogonal Frequency Division Multiplexing*) para o 5G NR. A modulação por multiplexação por divisão de frequência ortogonal (OFDM) foi escolhida como base para o 5G por uma série de razões.

Figura 11 – Pilha de protocolos da RAN 5G.



Fonte: O autor.

Ela tem eficiência espectral tanto no DL (*Download*) quanto no UL (*Upload*) para atender aos requisitos de altas taxas de dados. Além disso, a utilização fluente de MIMO e a combinação com *beamforming* fazem com que a OFDM contribua para compensar as perdas de propagação das ondas de rádio em bandas de alta frequência (ROMMER, 2020).

2.5.2.1 Sessão PDU

Uma das tarefas principais do sistema 5G é prover ao UE conectividade com a rede de dados. A rede de dados pode ser a internet, uma rede de dados específica de operadora ou uma rede de dados dedicada vertical como, por exemplo, de fábricas.

As sessões PDU podem ser de dois tipos principais: as sessões PDU do tipo IP e as sessões PDU do tipo ETH (*Ethernet*).

As sessões PDU do tipo IP oferecem um serviço de camada 3 incluindo a atribuição de endereços IP, filtragem e QoS baseada em fluxo IP. As sessões PDU do tipo IP são as sessões PDU mais comumente utilizadas para conectividade com a internet comum (CHANDRAMOULI; LIEBHART; PIRSKANEN, 2019).

A sessão PDU do tipo *Ethernet* oferece serviços de camada 2 incluindo controle de endereço MAC e/ou QoS baseado em VLAN (*Virtual LAN*), encaminhamento e filtragem de *frames*. As sessões PDU do tipo *Ethernet* são utilizadas nas aplicações industriais, em serviços de LAN (*Local Area Network*) usados em meio empresarial.

Quando o UE solicita o estabelecimento de uma Sessão de PDU, esta fornece um nome de rede de dados DNN (*Data Network Name*) que informa à rede principal 5G qual a DN que o UE deseja se conectar. Os DNN usados em uma rede podem ser específicos

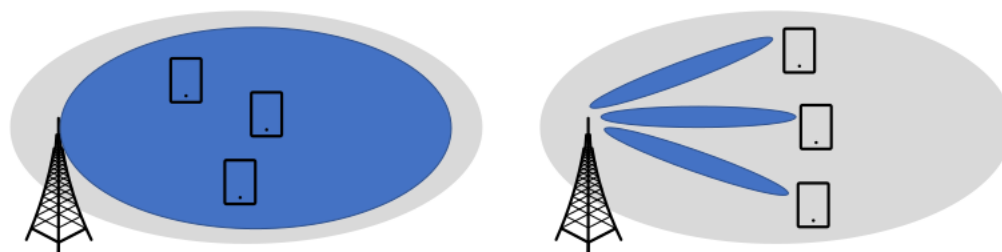
da operadora ou de uma rede privada (Intranet) (ROMMER, 2020).

2.5.2.2 Beamforming

Beamforming é uma técnica que permite direcionar a maioria da energia transmitida do transmissor para o receptor pretendido, ao invés de espalhar o sinal por toda a célula como ocorre no feixe único. Isso significa que o receptor recebe principalmente os sinais de rádio vindos da direção do transmissor, melhorando a relação sinal-ruído-mais-interferência, que é crucial para obter uma maior taxa de transferência de dados. É importante notar que, em uma implantação típica, a formação de feixe na direção de recepção é mais comum na estação base do que no dispositivo (ROMMER, 2020).

Já a técnica de feixe múltiplo implica em vários feixes de antena, cada um cobrindo uma parte menor da célula. Esses feixes são controláveis e direcionáveis dinamicamente, o que é utilizado para otimizar as características da conexão de rádio com cada dispositivo e maximizar o desempenho. A Figura 12 ilustra os conceitos de feixe único à esquerda, onde todo o sinal de rádio é disperso pela célula, e de multi-feixe à direita, onde o sinal é focalizado nos equipamentos do usuário.

Figura 12 – Feixe único e multi-feixe.



Fonte: (ROMMER, 2020).

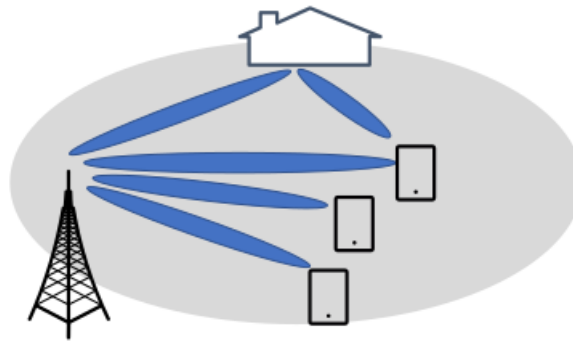
2.5.2.3 MIMO

O MIMO é uma tecnologia que utiliza múltiplas antenas de transmissão e recepção, o que permite criar canais separados com diferentes caminhos de propagação, possibilitando a transmissão de fluxos de dados entre o emissor e o receptor em uma mesma frequência. Ao receber os sinais, o receptor pode combinar ou selecionar o melhor deles para obter uma melhor qualidade do sinal recebido. Já o termo mMIMO (*Massive Multiple Input Multiple Output*) indica que a estação base utilizará uma quantidade muito maior do que oito antenas de transmissão e recepção (MAILER, C, 2020).

Quando uma única estação base se comunica com um único UE utilizando múltiplas antenas, é utilizado o SU-MIMO (*Single User Multiple-Input Multiple-Output*). Já no caso

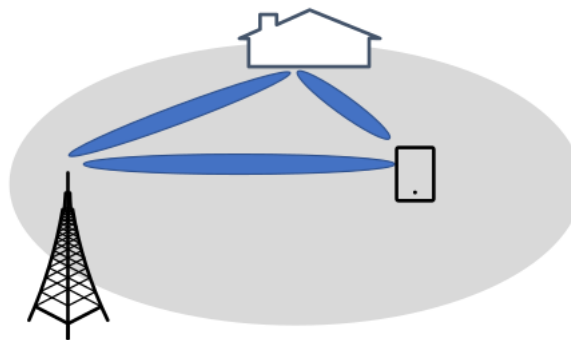
do MU-MIMO (*Multi User Multiple Input Multiple Output*), a estação base utiliza múltiplas antenas para se comunicar com múltiplos UEs simultaneamente (COX, 2020). As Figuras 13 e as 14 representam o funcionamento do UM-MIMO e do SU-MIMO respectivamente.

Figura 13 – UM-MIMO.



Fonte: (ROMMER, 2020).

Figura 14 – SU-MIMO.



Fonte: (ROMMER, 2020).

2.5.2.4 Fatiamento de Rede

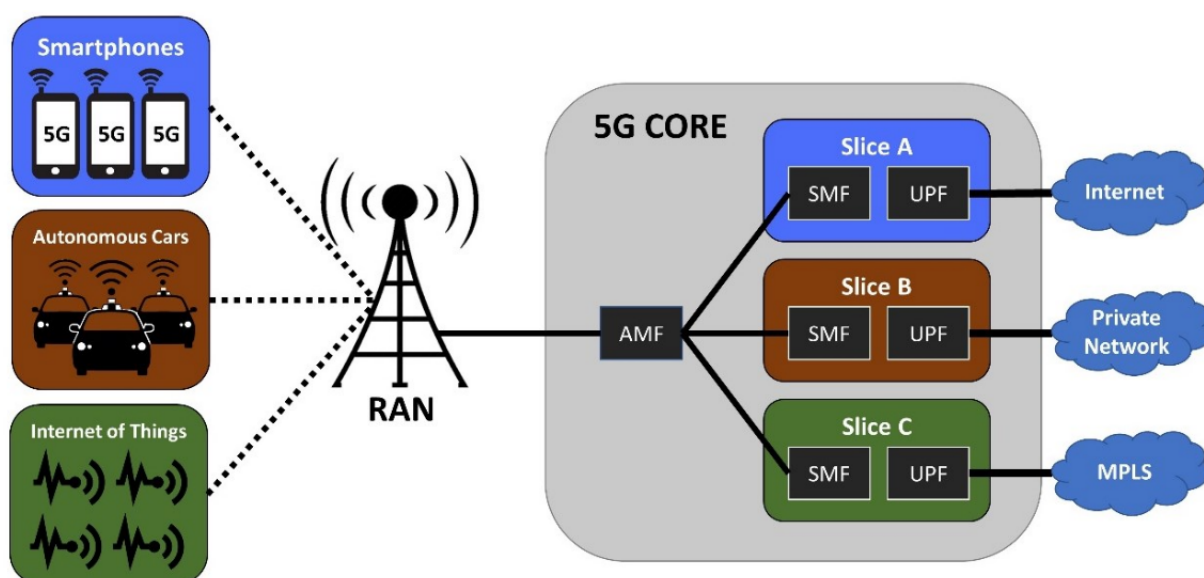
O fatiamento de rede, também denominado *network slicing*, é uma funcionalidade essencial na arquitetura do sistema 5G, particularmente nos modelos de negócios Rede como Serviço NaaS. Esta tecnologia envolve a segmentação da infraestrutura de rede física em múltiplas fatias lógicas, possibilitando o acesso diferenciado para cada tipo de dispositivo. Conseqüentemente, dispositivos que requerem comunicação em tempo real, como veículos autônomos, podem ser alocados em uma fatia com prioridade elevada e qualidade de serviço (QoS) superior, enquanto dispositivos menos dependentes de latência,

como *smartphones*, podem utilizar uma fatia de menor prioridade e custo (MAILER, C, 2020).

O fatiamento de rede também contribui para aprimorar a segurança das comunicações, uma vez que as redes lógicas distintas são mantidas isoladas entre si. Tecnologias como SDN (*Software-Defined Networks*) e NFV (*Network Functions Virtualization*) são cruciais para a implementação do fatiamento de rede.

A Figura 15 demonstra a aplicação do fatiamento de rede em três tipos distintos de conexões: smartphones, veículos autônomos e IoT (*Internet of Things*). É fundamental salientar que o fatiamento de rede constitui um recurso inovador do sistema 5G, desempenhando um papel crucial na elaboração de novos modelos de negócio baseados em serviços personalizados e na otimização da experiência do usuário (MAILER, C, 2020).

Figura 15 – Fatiamento de Rede.



Fonte: (MAILER, C, 2020).

Na arquitetura da rede 5G, cada fatia é reconhecida por um identificador conhecido como S-NSSAI (Single-Network Slice Selection Assistance Information). Esse identificador consiste em dois componentes: o SST (*Slice/Service Type*), que especifica o tipo de serviço ou segmento, e o SD (*Slice Differentiator*), que é um parâmetro opcional empregado para distinguir segmentos com o mesmo tipo de serviço (mesmo SST) (ROMMER, 2020).

Um dispositivo pode requisitar e utilizar múltiplos segmentos de rede simultaneamente. Ao processar uma solicitação, a rede utiliza o S-NSSAI para selecionar uma função de gerenciamento de acesso e mobilidade AMF adequada para atender o dispositivo inicialmente. A função de seleção de fatia de rede NSSF pode ser empregada para determinar o AMF apropriado de forma definitiva ou, na ausência do NSSF, o próprio AMF realizará a seleção (ROMMER, 2020).

2.5.3 Entendendo Contêineres, Docker e Kubernetes

O conceito de contêineres se tornou um pilar na computação moderna devido à sua promessa de interoperabilidade e consistência em diversos ambientes. Contêineres são uma forma de tecnologia de virtualização ao nível de sistema operacional que permite executar aplicações e suas dependências em recursos de processo isolados. Eles são leves e mais eficientes do que as máquinas virtuais tradicionais, pois não precisam carregar sistemas operacionais completos (CONTAINER. . . , 2023).

Nesse cenário, o Docker emergiu como a plataforma dominante para a criação e gerenciamento de contêineres. Docker é uma plataforma de código aberto que permite aos desenvolvedores e administradores de sistemas automatizar o empacotamento de aplicações e suas dependências em um contêiner. Isso torna os aplicativos não apenas portáteis entre diferentes plataformas e ambientes de nuvem, mas também consistentes em seu comportamento.

Entretanto, à medida que os aplicativos se tornam mais complexos e envolvem mais contêineres, surge a necessidade de uma ferramenta que gerencie e orquestre esses contêineres. Neste contexto é utilizado o Kubernetes, ou K8s.

Kubernetes é uma plataforma de código aberto que permite automatizar a implantação, o dimensionamento e o gerenciamento de aplicativos containerizados (KUBERNETES. . . , 2023). Ele agrupa contêineres que compõem uma aplicação em unidades lógicas para fácil gerenciamento e descoberta. Suas funcionalidades incluem escalonamento automático, distribuição de carga, automação de *rollouts* e *rollbacks*, visualização dos *logs* do *Pods* entre outros.

2.5.4 VirtualBox

A VirtualBox, também conhecida como Oracle VM VirtualBox, é uma ferramenta de software de virtualização que permite aos usuários executar vários sistemas operacionais em máquinas virtuais simultaneamente em um único computador físico. Ela é chamada de "*hipervisor* de tipo 2" ou "*hipervisor* hospedado", o que significa que é instalada como um aplicativo de software em um sistema operacional existente (VIRTUALBOX. . . , 2023).

A VirtualBox foi originalmente desenvolvida pela Innotek GmbH e, posteriormente, adquirida pela Sun Microsystems em 2008, que foi, por sua vez, comprada pela Oracle em 2010. Desde então, a VirtualBox tem sido mantida e desenvolvida pela Oracle (NUNES, 2015).

O uso da VirtualBox é bastante comum em cenários de testes e desenvolvimento. Por exemplo, os desenvolvedores podem querer testar seu software em diferentes sistemas operacionais sem a necessidade de ter vários computadores físicos. Da mesma forma, os administradores de sistemas podem usar a VirtualBox para criar máquinas virtuais com diferentes sistemas operacionais e configurações de redes para fins de testes ou para simular

diferentes ambientes.

Um dos principais benefícios da VirtualBox é sua portabilidade. Ela suporta uma grande variedade de sistemas operacionais hospedeiros, incluindo Windows, Linux, Mac e Solaris, entre outros.

Outra vantagem é que a VirtualBox é de código aberto e gratuita para uso pessoal e de avaliação. Isso a torna uma opção atrativa para realizar trabalhos.

2.5.5 UERANSIM

O UERANSIM é uma solução de software de código aberto, desenvolvida em C++, destinada à simulação de redes 5G. Esta ferramenta, que se destaca por oferecer uma implementação completa do UE e da gNB, permite a simulação de processos essenciais da rede 5G, tais como procedimentos de registro, estabelecimento de serviço, configuração de segurança, entre outros.

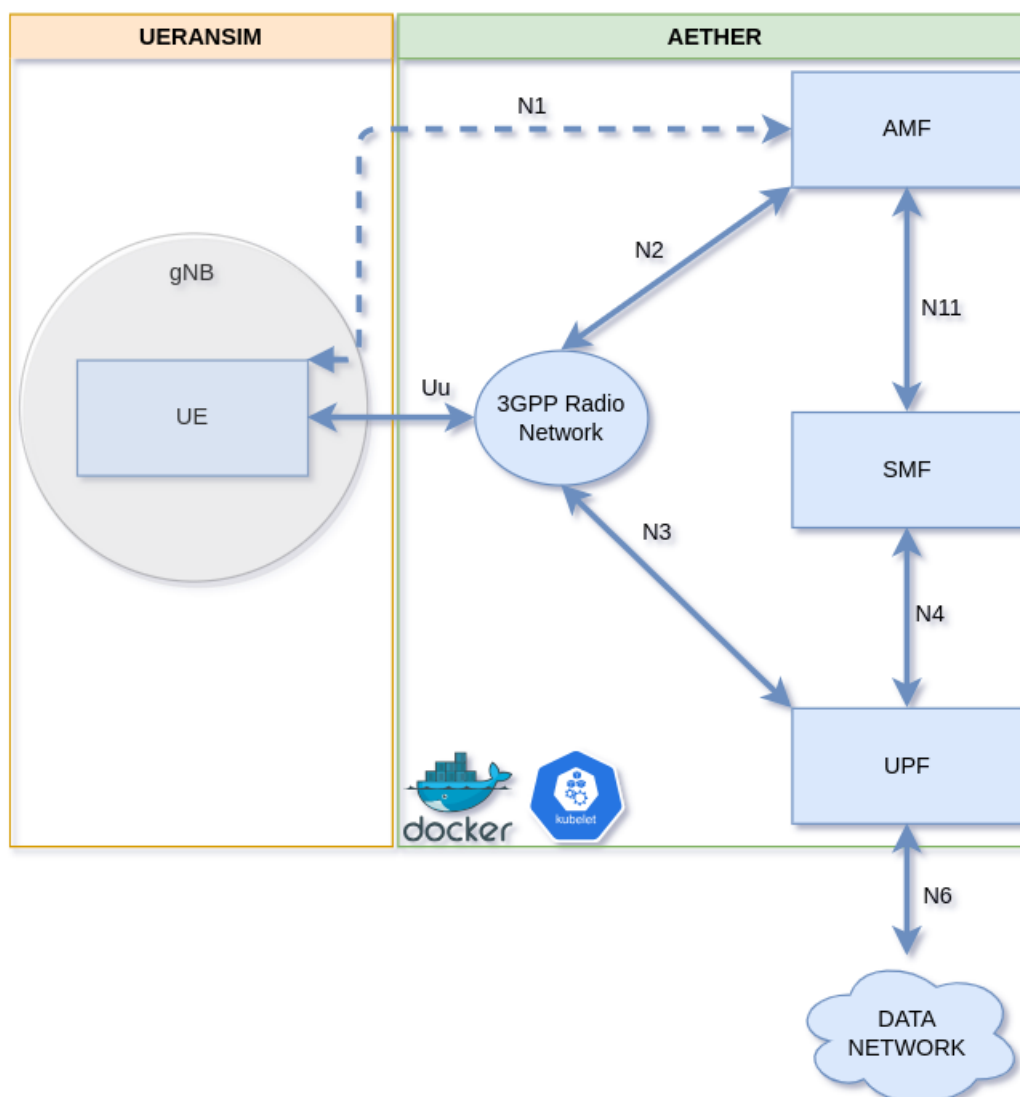
Essa plataforma de simulação fornece um ambiente controlado e reproduzível para experimentação e desenvolvimento da tecnologia 5G, sem a necessidade de acesso à infraestrutura real da rede 5G. Adicionalmente, seu caráter de código aberto permite que os usuários possam alterar e aprimorar o software conforme suas necessidades específicas, contribuindo para a evolução constante da ferramenta. Por isso, o UERANSIM se apresenta como um recurso valioso para pesquisadores, desenvolvedores e engenheiros que atuam com a tecnologia 5G (UERANSIM, 2023).

2.5.6 Aether

O Aether é uma plataforma de código aberto desenhada para gerir e facilitar a troca de dados entre sistemas de maneira segura e governada. Essa plataforma é composta por módulos como *Kernel*, *CKAN*, *Gather* e *Producer*, que respectivamente gerenciam metadados, fornecem um portal de dados abertos, coletam e publicam dados. Seu modelo baseado em esquemas assegura qualidade e consistência nos dados manipulados (AETHER..., 2023).

O Aether-in-a-Box é um projeto interno de pesquisa e desenvolvimento inserido no contexto do 5G. Ele busca fornecer uma solução completa para testes e simulações de tecnologias utilizando o núcleo do 5G, abrangendo desde equipamentos de rede até aplicações (AETHER PROJECT, Acesso em: 23 jun. 2023). A ideia é criar um ambiente controlado e flexível com todos os elementos do núcleo do 5G, permitindo desenvolver simulações completas dos recursos do 5G em redes privadas, sendo possível também utilizar UE reais em conjunto.

Figura 16 – Sistema UERANSIM-Aether.



Fonte: O autor.

2.5.7 SD-WAN

Rede de Longa Distância Definida por Software (SD-WAN), é uma tecnologia de rede que simplifica a administração e operação de uma rede de longa distância (WAN).

Uma WAN é uma rede que conecta dispositivos ou outras redes que estão distantes entre si, geralmente em diferentes locais. Geralmente as WANs são utilizadas por empresas para conectar suas sedes a filiais, data centers e serviços na nuvem.

A parte de "SD" é a sigla para "Software Defined", ou "Definido por Software", em tradução direta. Tradicionalmente, a administração de uma rede era fortemente dependente do hardware e exigia a configuração manual de roteadores e outros dispositivos. Com o SD-WAN, a administração da rede é abstraída do hardware e gerida por um software. Isso permite que a rede seja configurada de maneira otimizada e seu monitoramento seja feito

de maneira centralizada e automática (SEGEČ *et al.*, 2021).

Com esse roteamento inteligente, é possível escolher o melhor caminho para o tráfego de dados com base nas condições da rede em tempo real, e a capacidade de utilizar simultaneamente várias conexões de rede, como banda larga, conexões via satélite e 5G.

A interseção do SD-WAN e do 5G abre um leque de possibilidades para gerenciar e otimizar redes. À medida que mais dispositivos se tornam capazes de se conectar via 5G, o tráfego de rede está propenso a aumentar substancialmente. Nesse contexto o SD-WAN pode ser usada para gerenciar eficientemente esse aumento de tráfego de rede, garantindo que as redes não fiquem congestionadas e o desempenho não seja comprometido.

A flexibilidade e a gestão centralizada da SD-WAN podem facilitar o gerenciamento de redes 5G. Além disso, a capacidade da SD-WAN de utilizar várias conexões simultaneamente podem permitir que os administradores de rede combinem conexões 5G 3GPP e NON3GPP para maximizar o desempenho e a confiabilidade da rede.

2.5.8 5G no meio industrial e IoT

A quinta geração de redes móveis, conhecida como tecnologia 5G, promete ser uma força revolucionária. Com a capacidade de transformar a indústria através da Indústria 4.0, ela se destaca pela sua velocidade excepcional e alta capacidade, permitindo a transmissão ágil de enormes volumes de dados, uma exigência fundamental para as aplicações industriais contemporâneas.

A interconexão física entre componentes de uma rede industrial normalmente é feita via *Ethernet*. Contudo, existem soluções sem fio, como *WirelessHart*, *Zigbee* e *Bluetooth*, que não são universais e enfrentam desafios ausentes no 5G, como questões de confiabilidade, latência, segurança e robustez (MAILER, C, 2022).

A velocidade impressionante do 5G, combinada com a latência extremamente baixa, torna esta tecnologia ideal para o controle de máquinas em tempo real e sistemas autônomos.

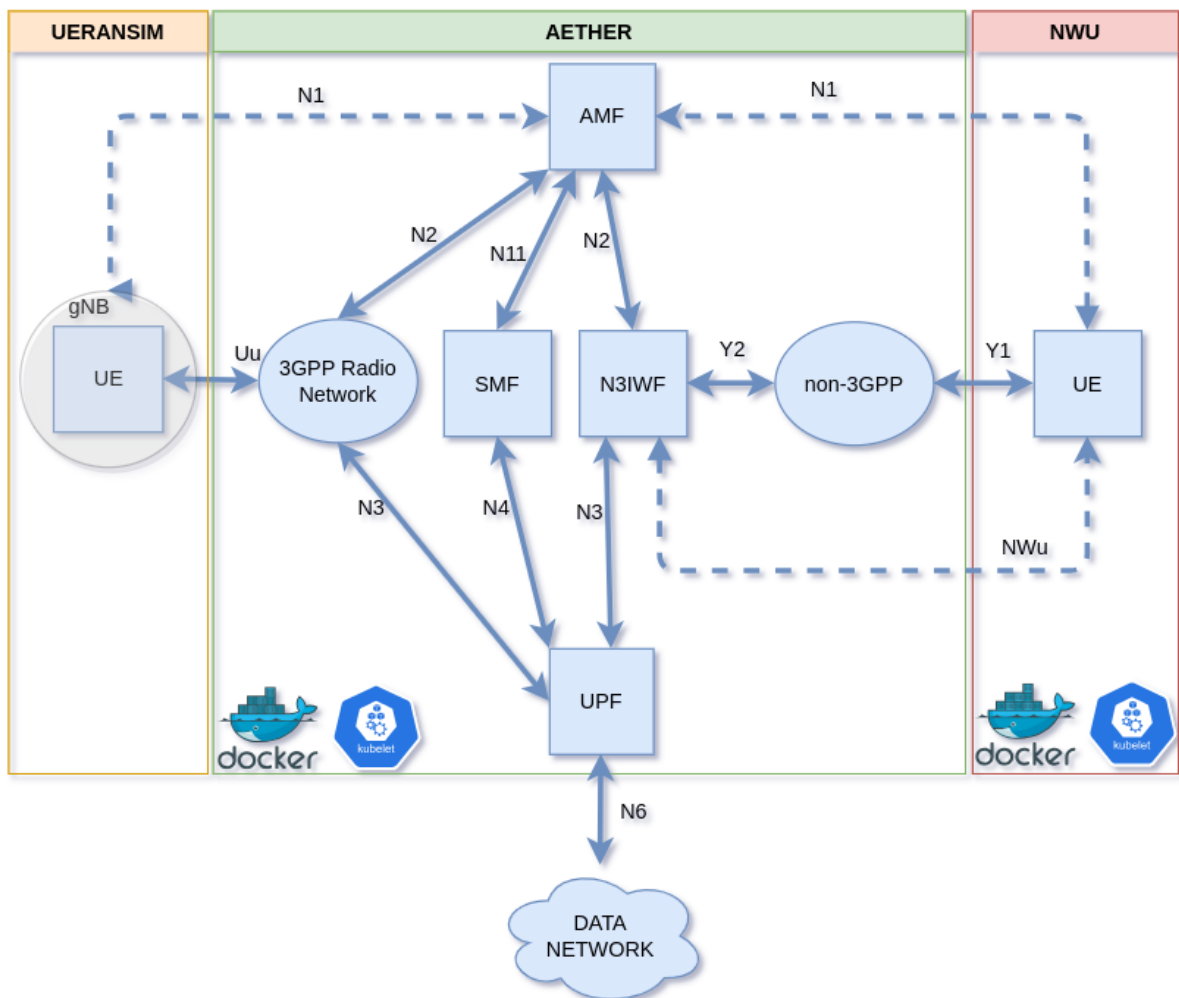
No contexto da Internet das Coisas Industrial (IIoT), o 5G se destaca ao suportar múltiplos dispositivos conectados simultaneamente. Em uma indústria onde uma crescente quantidade de sensores e dispositivos precisa se comunicar, essa capacidade de gerenciar conexões massivas de máquinas é de valor inestimável.

No entanto, um desafio a ser vencido é a integração de dispositivos industriais mais antigos com os novos que utilizam 5G, uma vez que substituir todos os dispositivos por novos compatíveis com 5G é economicamente inviável. Uma solução para este problema é estabelecer uma conexão entre os dispositivos 5G e não 5G através de uma conexão NON3GPP.

3 SISTEMA PROPOSTO

Neste capítulo, é apresentado o sistema proposto para a implementação e teste de conectividade 3GPP e NON3GPP com um núcleo de rede 5G. O sistema foi projetado para estabelecer uma conexão tanto 3GPP quanto NON3GPP com o núcleo do 5G, além de possibilitar a comunicação entre dois UEs, cada uma utilizando um destas tecnologias.

Figura 17 – Sistema completo.



Fonte: O autor.

Como podemos observar na Figura 17, à esquerda, utilizamos a ferramenta de simulação UERANSIM para emular um UE e um gNodeB, possibilitando a conexão 3GPP com o núcleo 5G. À direita, estabelecemos uma conexão NON3GPP por meio de um túnel *NWu*.

O objetivo principal dessa configuração é permitir que ambos os UEs, 3GPP e NON3GPP, se comuniquem de forma eficiente através do núcleo da rede 5G. Além disso, a configuração foi projetada para possibilitar a comunicação direta entre os dois UEs,

independentemente do tipo de conexão que elas estão utilizando.

A conexão entre UE e o Aether, estabelecida através da interface NWu, é designada como UE-NON3GPP neste trabalho. Neste contexto, a interface NWu age como um canal de comunicação, conectando a UE ao Aether e facilitando a troca de dados e o controle na rede 5G, como ilustrado na Figura 4.

Esta conexão permite que a UE aproveite as funcionalidades avançadas e os recursos da rede 5G, como altas velocidades de transferência de dados, baixa latência, conectividade simultânea a várias redes e suporte a uma variedade de serviços.

A conexão entre a UE e o Aether, estabelecida através do UERANSIM, é referenciada como UE-gNB, representando uma conexão 3GPP. Isto nos permite simular uma conexão nativa 5G, emulando as interações entre o dispositivo do usuário e as funções do núcleo da rede, conforme retratado na Figura 16. Com isso, podemos realizar testes e análises detalhadas, verificar a correta implementação dos protocolos de comunicação e avaliar o desempenho da rede em diferentes condições.

Com esta configuração, nosso objetivo é avaliar a funcionalidade e eficiência das conexões 3GPP e NON3GPP em um ambiente de rede 5G e examinar o comportamento e desempenho quando ambas as UEs tentam se comunicar entre si.

3.1 PREPARAÇÃO DO AMBIENTE

Primeiramente, foi necessário preparar o ambiente de teste. Para isso, foram utilizadas três máquinas virtuais rodando Ubuntu 18.04 LTS

Para criar três máquinas virtuais usando o VirtualBox, cada uma executando o Ubuntu 18.04, o primeiro passo foi baixar a imagem ISO do Ubuntu 18.04 (UBUNTU, Acesso em: 23 jun. 2023). A imagem ISO utilizada foi obtida no site oficial do Ubuntu e optado pela versão LTS (*Long Term Support*), que oferece suporte por um período mais longo.

Após obter a imagem ISO do Ubuntu, foi realizado a instalação do VirtualBox utilizando os seguintes comandos mostrados no Código 3.1.

```
1 $ sudo apt-get update
2 $ sudo apt-get install virtualbox
```

3.1 – Instalação do VirtualBox.

Em seguida, foram criadas três máquinas virtuais distintas para executar o Aether, UE-NON3GPP e UE-gNB. Cada uma dessas máquinas virtuais teve uma alocação de memória RAM específica - 10GB para Aether e 1GB para UE-NON3GPP e UE-gNB. Devido ao Aether ser todo o núcleo e gerenciamento do 5G requer uma grande quantidade de memória RAM.

Na criação das máquinas virtuais, foram criados discos rígidos virtuais específicos para cada uma. As maquinas foram configurados como *VirtualBox Disk Image*, uma

opção que permite a alocação dinâmica de espaço, economizando assim o espaço do disco rígido do computador host. A capacidade de cada disco rígido foi ajustada conforme a necessidade de cada software, sendo 50GB para Aether, 10GB para UE-NON3GPP e 10GB para UE-gNB.

Em relação à quantidade de núcleos de CPU alocados, foi definida uma distribuição de 4 núcleos para Aether e 1 núcleo para UE-NON3GPP e UE-gNB.

No que diz respeito à configuração da rede, cada máquina virtual foi configurada com dois adaptadores de rede, um do tipo NAT e outro do tipo *Host-Only Adapter*.

É importante mencionar o comando `sudo` que durante todo o desenvolvimento do trabalho foi utilizado para a execução de comandos com privilégios de superusuário. Este comando permitiu a execução de várias ações que requeriam permissões elevadas.

Após a instalação do Ubuntu nas VMs foram realizadas atualizações utilizando os comandos “update” e “upgrade”, que asseguram que o sistema operacional esteja atualizado com as últimas correções e melhorias. Para confirmar todas as mudanças realizadas e garantir o correto funcionamento do sistema, um processo de reinicialização foi realizado após cada atualização como pode ser visto no Código 3.2

```
1 $ sudo apt update
2 $ sudo apt upgrade
3 $ sudo reboot
```

3.2 – Atualização do ambiente.

Foi executado o comando “ping” para verificar se a VM esta conectada a internet, enviando pacotes ICMP para um endereço IP na rede, neste caso, google.com. E o comando ‘ip addr show’ é usado para exibir informações detalhadas sobre os endereços de rede (IP) configurados em todas as interfaces de rede do seu sistema. O Ping mostra informações como endereços IPv4 e IPv6, máscaras de sub-rede, status das interfaces e outras configurações de rede. Contudo, para utilizar o “ip addr show”, é necessário instalar o pacote `iproute2` previamente, conforme mostrado no Código 3.3.

```
1 $ ping google.com
2 $ sudo apt update
3 $ sudo apt install iproute2
4 $ ip addr show
```

3.3 – Preparando o ambiente.

Os *hostnames* das VMs foram alterados para melhor organizar o ambiente de desenvolvimento e garantir uma nomenclatura mais clara e intuitiva, podemos ver no Código 3.4 a Alteração do Aether.

```
1 # Alterando o hostname da VM1 para "aether"  
2 $ sudo hostnamectl set-hostname aether
```

3.4 – Alteração do hostname.

O nome de domínio das três máquinas virtuais foi alterado para estar conforme os novos nomes, como exemplificado em 3.5. É importante ressaltar que esta configuração foi realizada para o endereço IP 127.0.0.1, também conhecido como endereço de *loopback*. Esse endereço IP permite que o *host* direcione tráfego para si.

```
1 $ sudo vi/etc/hosts  
2 127.0.0.1 localhost  
3 127.0.1.1 aether
```

3.5 – Configuração de Loopback.

No próximo passo, foram atribuídos endereços IPs estáticos à interface de rede *Host-Only Adapter* (enp0s3), e a configuração utilizando o protocolo DHCP (*Dynamic Host Configuration Protocol*) foi desativada nessa mesma interface, conforme demonstrado para a VM do Aether. Após a definição do endereço IP estático na VM, os comandos “netplan try” e “netplan apply” foram empregados para verificar a validade do endereço IP e para aplicar as configurações. Para confirmar as alterações realizadas, um processo de reinicialização foi efetuado, como apresentado no Código 3.7.

```
1 $ sudo vi /etc/netplan/00-installer-config.yaml  
2  
3 network :  
4   ethernets :  
5     enp0s3 :  
6       dhcp4 : true  
7     enp0s8 :  
8       dhcp4 : false  
9       addresses : [192.168.56.101/24]  
10  version : 2
```

3.6 – Configuração do netplan.

```
1 $ sudo netplan try  
2 $ sudo netplan apply  
3 $ ip addr show  
4 $ sudo reboot
```

3.7 – Aplicação do netplan.

Sabendo o IP das VMs é possível utilizar o protocolo SSH (*Secure Shell*) para acessar remotamente as VMs e facilitar visibilidade e a operacionalidade, juntamente a isso foi instalado o Guake por que é um terminal que permite múltiplas abas abertas na mesma tela, auxiliando na visualização, o Código pode ser visto em ??.

3.2 INSTALAÇÃO DO AETHER

Na primeira VM, foi realizado a instalação do Aether-in-a-box O procedimento seguiu os passos descritos na documentação oficial do Aether (AETHER PROJECT, Acesso em: 23 jun. 2023).

Para a inicial configuração do ambiente de desenvolvimento no sistema operacional Ubuntu, foram necessárias várias etapas. Primeiramente, utilizou-se o comando “sudo apt install git make”, a fim de instalar dois pacotes de software essenciais: o Git e o Make.

O Git é reconhecido como um sistema de controle de versão distribuído, cuja aplicação é preponderante em diversas situações, devido à sua habilidade de rastrear alterações em arquivos de código-fonte e sua eficácia em facilitar a colaboração em projetos de desenvolvimento de software.

No entanto, o Make apresenta um papel distinto, mas não menos importante. Trata-se de uma ferramenta destinada à automação de compilação que, por sua vez, interpreta um arquivo intitulado Makefile. Este arquivo contém instruções detalhadas para compilar e vincular programas a partir dos arquivos de código-fonte.

Após a conclusão da instalação desses pacotes, foi empregado o comando “git clone” para clonar o repositório do Aether. Esta operação produziu uma cópia integral do repositório especificado, situada no diretório atual.

Estes comandos, executados em sequência, se mostraram imprescindíveis para a adequada configuração do ambiente e para a preparação da VM do Aether. Isso permitiu que a VM estivesse apta para futuras atividades relacionadas ao desenvolvimento do trabalho em questão.

```
1 $ sudo apt install git make
2 $ git clone "https://gerrit.opencord.org/aether-in-a-box"
3 $ cd aether-in-a-box
```

3.8 – Instalando o Aether

O próximo passo foi alterar o arquivo sd-núcleo-5g-values.yaml, foi preciso realizar uma alteração no repositório para que pudéssemos utilizar as versões modificadas do AMF e do AUSF disponíveis em outro repositório, como pode ser visto no Código 3.9. Esse procedimento será detalhado na seção 3.6.

```

1 $ vi sd-n cleo -5g-values.yaml
2
3 5g-control-plane:
4   enable5G: true
5   images:
6     # repository: "registry.opennetworking.org/docker.io/"
7     tags:
8       amf: amf:latest
9       ausf: ausf:latest

```

3.9 – Configuração dos tags do AMF e AUSF.

Para expor IP externo e porta do serviço AMF para fazer a conexão com o UERANSIM é necessário adicionar esse trecho de Código 3.10, sendo o protocolo SCTP (*Stream Control Transmission Protocol*) com número de porta padrão 38412 é usado entre gNB e AMF e o IP da VM do Aether.

```

1 # O protocolo SCTP com número de porta padrão 38412
   usado entre gNB e AMF.
2
3   amf:
4     # use externalIP if you need to access your AMF from
   remote setup and you don't
5     # want setup NodePort Service Type
6     ngapp:
7       externalIp: "192.168.56.101"
8       port: 38412

```

3.10 – Configuração da porta 38412.

Também foi necessário adicionar IMSI para os cartões SIM simulados, com a chave de transporte e os valores OPC, importante mencionar que esses mesmos valores deverão ser utilizados nos UEs para a conexão, como pode ser visto no Código 3.11.

```

1     subscribers:
2       - ueId-start: "315010999912301"
3         ueId-end: "315010999912303"
4         plmnId: "315010"
5         opc: "69d5c2eb2e2e624750541d3bbc692ba5"
6         op: ""
7         key: "000102030405060708090a0b0c0d0e0f"

```

3.11 – Configuração dos parâmetros do UE.

1

3.3 INSTALAÇÃO DO UERANSIM

O simulador UERANSIM, obtido a partir do repositório oficial disponível em (UERANSIM, 2023), foi clonado e as suas aplicações essenciais foram instaladas, pode ser visto no Código 3.12. Esse simulador possui certos arquivos de configuração destinados a se integrar com aplicações de núcleo do 5G (Aether).

Primeiramente, o arquivo de configurações associado ao simulador da Antena 5G foi modificado conforme o código mostrado na próxima página. O endereço IP estático da VM do UE-gNB foi estabelecido como o endereço IP da simulação do link de rádio dentro do UERANSIM. Neste mesmo arquivo foi adicionado o endereço IP estático da VM do Aether como o endereço do AMF, um melhor esclarecimento da conexão pode ser visto na Figura 18.

```
1 $ git clone https://github.com/aligungr/UERANSIM
2 $ sudo apt update
3 $ sudo apt upgrade
4 $ sudo apt install make
5 $ sudo apt install gcc
6 $ sudo apt install g++
7 $ sudo apt install libsctp-dev lksctp-tools
8 $ sudo apt install iproute2
9 $ sudo snap install cmake --classic
10 $ cd ~/UERANSIM
11 $ make
```

3.12 – Instalação do UERANSIM.

A configuração da estação base, gNB, no âmbito do UERANSIM demandou uma definição precisa dos parâmetros de acordo com as configurações previamente estabelecidas no Aether. O trecho de Código 3.13 ilustra este processo.

```
1 mcc: '208'           # Mobile Country Code value
2 mnc: '93'           # Mobile Network Code value (2 or 3
  digits)
3
4 nci: '0x000000010'  # NR Cell Identity (36-bit)
```

¹ O código completo de `sd-núcleo-5g-values.yaml` pode ser encontrado no Apêndice A.

```

5 idLength: 32          # NR gNB ID length in bits [22...32]
6 tac: 1              # Tracking Area Code
7
8 linkIp: 192.168.56.102 # gNB's local IP address for Radio
   Link Simulation (Usually same with local IP)
9 ngapIp: 192.168.56.102 # gNB's local IP address for N2
   Interface (Usually same with local IP)
10 gtpIp: 192.168.56.102 # gNB's local IP address for N3
   Interface (Usually same with local IP)
11
12 # List of AMF address information
13 amfConfigs:
14   - address: 192.168.56.101
15     port: 38412
16
17 # List of supported S-NSSAIs by this gNB
18 slices:
19   - sst: 0x1
20     sd: 0x010203
21
22 # Indicates whether or not SCTP stream number errors should
   be ignored.
23 ignoreStreamIds: true

```

3.13 – Configuração do gNB

O mesmo procedimento é aplicado para as configurações do UE, como pode ser visto no Código 3.14.

```

1 # IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15
   digits)
2 supi: 'imsi-208930100007501'
3 # Mobile Country Code value of HPLMN
4 mcc: '208'
5 # Mobile Network Code value of HPLMN (2 or 3 digits)
6 mnc: '93'
7 # Routing Indicator
8 routingIndicator: '0000'
9 # Permanent subscription key
10 key: '5122250214c33e723a5dd523fc145fc0'
11 # Operator code (OP or OPC) of the UE

```

```
12 op: '981d464c7c52eb6e5036234984ad0bcf'
13 # This value specifies the OP type and it can be either 'OP'
    or 'OPC'
14 opType: 'OPC'
15 ...
16 # List of gNB IP addresses for Radio Link Simulation
17 gnbSearchList:
18   - 192.168.56.102
19 ....
20 # Supported integrity algorithms by this UE
21 integrity:
22   IA1: true
23   IA2: false
24   IA3: false
25 # Supported encryption algorithms by this UE
26 ciphering:
27   EA1: false
28   EA2: false
29   EA3: false
```

3.14 – Configuração do UE

2

Para garantir o encaminhamento adequado do tráfego pela interface correta, foi necessário adicionar uma nova rota na tabela de roteamento. No trecho de Código 3.16 podemos observar o comando utilizado para adicionar a rota.

```
1 $ ip route add 192.168.252.3 via 192.168.56.101
```

3.15 – Adição de rota.

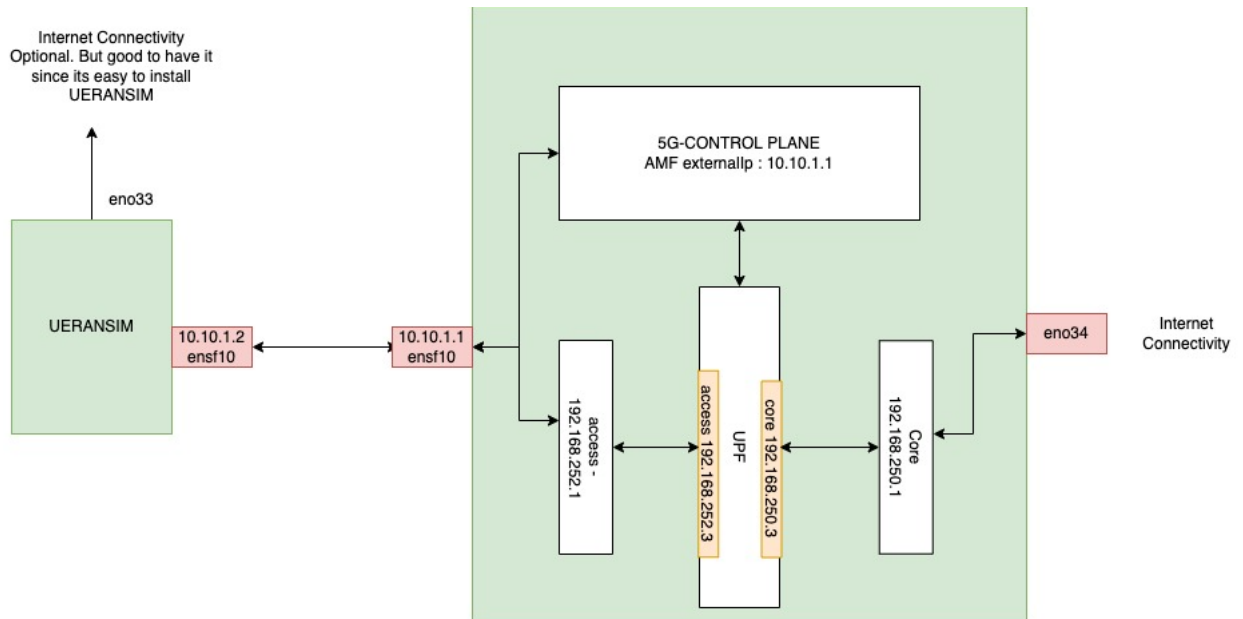
Nesse caso, o endereço de rede 192.168.252.3 foi configurado para ser encaminhado através do IP 192.168.56.101, que corresponde ao IP do Aether.

Após adicionar a rota, foi verificado a tabela de roteamento utilizando o comando “ip addr show” para garantir que a rota foi configurada corretamente. Essa etapa foi importante para assegurar que o tráfego seria direcionado adequadamente pela interface desejada.

A Figura 18 ilustra a interação entre o UERANSIM e o Aether.

² O código completo de configuração do gNB e do UE podem ser encontrados no Apêndice B.

Figura 18 – Conexão entre UERANSIM e Aether.



Fonte: (AETHER. . . , 2023)

Os comandos representados no trecho de Código 3.16 são empregados para ativar a estação base gNB e o UE, ambos do UERANSIM. A verificação do funcionamento correto pode ser realizada enviando pacotes de “ping” para o endereço IP 8.8.8.8 através da interface uesimtun0.

```
1 $ sudo build/nr-gnb -c config/aether-gnb.yaml
2 $ sudo build/nr-ue -c config/aether-ue.yaml
3 $ ping -I uesimtun0 8.8.8.8
```

3.16 – Ativação da gNB e do UE.

3.4 CONFIGURAÇÃO DO N3IWF USANDO O PROJETO FREE5GC-COMPOSE

Para configurar o N3IWF na solução, foi utilizado o projeto free5gc-compose, que fornece uma estrutura de composição para implantação e execução do N3IWF (FREE5GC, Acesso em: 23 jun. 2023[d]).

Foi utilizado o comando “git clone” para copiar o repositório e “docker-compose up” para iniciar a execução dos serviços definidos no arquivo de composição do Docker, responsáveis pela implantação e configuração do N3IWF.

Esses comandos permitem a configuração adequada do N3IWF, garantindo a correta integração com o restante da rede e possibilitando o funcionamento adequado da solução desenvolvida.

3.5 CONFIGURANDO O NWU

Para configurar o *NWu* foi utilizado o repositório (FERNANDO AUGUSTO SILVA FERRAZ, Acesso em: 23 jun. 2023).

Após clonar o repositório foi realizado o *build* da imagem Docker do *NWu* com o comando “docker build” e executado o *NWu* em um contêiner Docker utilizando o seguinte comando:

```
1 $ sudo docker run --network host --privileged -it --rm nwu -a
   internet -d 192.168.56.101 -M 208 -N 93 -I
   208930100007501 -K 5122250214c33e723a5dd523fc145fc0 -s
   192.168.56.102 -C 981d464c7c52eb6e5036234984ad0bcf --sst=1
```

3.17 – Execução do NWu.

Esse comando apresentado no trecho de Código 3.17 executa o *NWu* em modo interativo e configura os parâmetros necessários para estabelecer a conexão com o núcleo da rede 5G, de forma semelhante feita no UE-gNB.

3.6 ALTERAÇÃO E ADAPTAÇÃO DO AMF E DO AUSF

Para garantir o correto funcionamento da solução, foi necessário realizar alterações e adaptações nos scripts do AMF e AUSF, resultando na criação de uma nova imagem Docker.

Inicialmente, os *scripts* do AMF e AUSF foram obtidos a partir do repositório do projeto free5gc (FREE5GC, Acesso em: 23 jun. 2023[c]). Em seguida, for “diff” para ver as alterações e depois aplicá-las. Para o AUSF foi utilizado o comando “diff” dos arquivos 3, 4 e 5 do repositório (FREE5GC, Acesso em: 23 jun. 2023[a]).

Já para o AMF foi utilizado o comando “diff” dos arquivos 35, 38, 41, 42, 43, 45 e 50 do repositório (FREE5GC, Acesso em: 23 jun. 2023[b]).

Essas modificações e adaptações foram essenciais para garantir a compatibilidade e o correto funcionamento do AMF e AUSF na solução desenvolvida.

Após as alterações realizadas no AMF e AUSF, foi necessário realizar o *build* novamente desses componentes. O processo de *build* consiste em compilar o código-fonte e gerar uma nova imagem Docker com as modificações aplicadas.

Os seguintes comandos utilizados para realizar o *build* do AMF estão apresentados no trecho de Código 3.18

```
1 $ docker build -t amf .
2 $ docker save -o ~/images/new_amf.tar amf
```

```
3 $ sudo ctr -a /run/k3s/containerd/containerd.sock -n k8s.io  
   image import ~/images/new_amf.tar
```

3.18 – Build do novo AMF.

Iniciou-se o processo com o comando “`docker build -t amf .`”, que cria uma nova imagem Docker nomeada “`amf`” a partir do Dockerfile existente no repositório do AMF modificado. Este comando efetua a compilação do código-fonte e gera a imagem Docker com as modificações aplicadas. Em seguida, o comando “`docker save -o /images/new_amf.tar amf`” é utilizado para salvar a imagem recém-criada do AMF em um arquivo tar denominado “`new_amf.tar`”.

Posteriormente, com o comando “`sudo ctr -a /run/k3s/containerd/containerd.sock -n k8s.io image import /images/new_amf.tar`”, a imagem do AMF é importada para o ambiente Kubernetes. Para isso, utiliza-se o utilitário `ctr` para conectar-se ao `containerd` e realizar a importação.

Os mesmos comandos foram empregados para o AUSF. Estes passos são cruciais para assegurar que as alterações realizadas tanto no AMF quanto no AUSF sejam adequadamente refletidas na implementação final da solução.

3.7 FERRAMENTAS

Com o auxílio do Ping, Tcpdump, Wireshark e Kubectl, foi possível realizar testes, capturar e analisar o tráfego de rede, além de implantar e gerenciar aplicativos. Essas atividades foram fundamentais para o *debug* e para entender o comportamento da rede, identificando possíveis problemas, anomalias e garantindo um melhor entendimento do sistema como um todo. Através dessas ferramentas, foi possível realizar uma análise detalhada do tráfego, monitorar o estado dos serviços e obter informações cruciais para o desenvolvimento e aprimoramento do projeto.

- Ping: Foi utilizada a ferramenta Ping para verificar a conectividade entre dispositivos na rede, testando a disponibilidade e latência de comunicação;
- Tcpdump: Foi utilizado o Tcpdump para capturar e analisar o tráfego de rede em tempo real. Essa ferramenta permitiu examinar os pacotes que transitavam pela rede e identificar possíveis problemas ou anomalias;
- Wireshark: Além do Tcpdump, também utilizei o Wireshark para análise detalhada do tráfego de rede. O Wireshark proporcionou uma visualização mais abrangente dos pacotes capturados, permitindo uma análise mais profunda dos protocolos e das Foi utilizado contidas nos pacotes;
- Kubectl: Utilizei o Kubectl para implantar e gerenciar aplicativos em um ambiente Kubernetes. Essa ferramenta foi essencial para realizar implantações,

monitorar o estado dos serviços, *logs* e depurar possíveis problemas na infraestrutura.

- Netperf: Foi utilizado o Netperf para medir a taxa de transferência, latência e outras métricas importantes. Suportando diversos protocolos e oferecendo vários tipos de testes para otimizar e diagnosticar problemas de desempenho na rede.

3.8 METODOLOGIA DOS TESTES

Para assegurar que a conexão UE-gNB navegasse através do núcleo 5G, a estratégia empregada foi utilizar a interface previamente criada durante a conexão entre o UERAN-SIM e o Aether. A interface 'uesimtun0' é destacada no trecho de Código 3.19

```
1 $ sudo ping 192.168.250.125 -I uesimtun0 -c 5000 -i 0.01
```

3.19 – Ping pela interface uesimtun0

Ao executar o comando, o destino para o teste de “ping” é definido como o endereço IP 192.168.250.125.

Para o teste, é especificada a interface de rede uesimtun0 por meio do parâmetro -I. Além, são definidos outros parâmetros, como -c 5000, que indica o número de pacotes de “ping” a serem enviados (neste caso, 5000 pacotes), e -i 0.01, que define o intervalo de tempo entre o envio de cada pacote (0.01 segundos, ou seja, um pacote a cada 10 milissegundos).

E para a segunda parte dos testes foi utilizado o “netperf”, para capturar os valores de largura de banda.

No caso da UE-NON3GPP, não dispúnhamos da facilidade de utilizar a interface diretamente. Portanto, para garantir que o tráfego fosse roteado através do núcleo 5G até o Aether durante nossos experimentos, foi necessária a criação de uma interface 'dummy' dentro da VM do Aether. Esta interface foi atribuída ao endereço IP 10.0.0.1.

```
1 $ netperf -H 10.0.0.1 -l 60 -t TCP_STREAM
```

3.20 – Netperf TCP STREAM

No comando apresentando no trecho de Código 3.20 é executado utilizando o protocolo TCP (*Transmission Control Protocol*) para medir a taxa de transferência de um fluxo de dados. O *host* de destino especificado é o endereço IP 10.0.0.1, e o teste é executado por 60 segundos.

```
1 $ netperf -H 10.0.0.1 -l 60 -t TCP_RR
```

3.21 – Netperf TCP RR

No outro comando apresentando no trecho de Código 3.21 é utilizado o teste de solicitação/resposta TCP, que mede o número de transações TCP por segundo. Também é especificado o host de destino como 10.0.0.1 e a duração do teste de 60 segundos.

Além disso, a ferramenta Htop foi utilizada para monitorar o uso de CPU e memória RAM em cada máquina virtual, possibilitando a identificação de possíveis problemas de recursos durante os testes.

Esses comandos e ferramentas são úteis para avaliar o desempenho da rede, medir a taxa de transferência e identificar possíveis gargalos ou problemas de recursos nas máquinas envolvidas nos testes.

4 RESULTADOS

Neste capítulo, apresentaremos os resultados coletados por meio de imagens, gráficos, códigos-fonte e tabelas, visando oferecer uma compreensão mais clara acerca dos resultados da simulação.

É importante mencionar que, quando a conexão é estabelecida entre uma UE e o Aether, o núcleo do 5G atribui um novo endereço de IP ao UE para ser responsável por enviar e receber os pacotes. Essa nova interface pode ser visualizada dentro da UE utilizando o comando 'ip a' após a conexão estar estabelecida.

Por exemplo, ao executar o comando “ip r show” dentro da UE-NON3GPP, serão listadas as interfaces de rede disponíveis, incluindo o novo IP atribuído pelo núcleo do 5G. Essa informação é relevante para entendermos como a comunicação entre o UE e o Aether é estabelecida e como os pacotes são roteados corretamente pela rede.

```

1 ubuntu@UE-NON3GPP:~$ ip r show
2
3 67: tun3: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500
   qdisc fq_codel state UNKNOWN group default qlen 500
4   link/none
5   inet 172.250.237.125/32 scope global tun3
6       valid_lft forever preferred_lft forever
7   inet6 fe80::93e9:2594:d265:8d3f/64 scope link stable-
   privacy
8       valid_lft forever preferred_lft forever

```

4.1 – Lista das interfaces de rede.

Após estabelecer toda a arquitetura do trabalho, o primeiro teste realizado foi o de “ping”. Foi realizado o teste de conectividade entre os UEs, ou seja, o UE-NON3GPP realizou o “ping” no UE-gNB e vice-versa. Essa etapa permitiu verificar se todo o caminho de conexão estava funcionando corretamente, onde o IP 172.250.237.125 é o IP atribuído pelo núcleo a UE-NON3GPP e o IP 172.250.237.124 é o IP atribuído pelo núcleo a UE-gNB.

```

1 $ ping 172.250.237.124

```

4.2 – Ping para o UE

Após utilizar o “ping”, foi possível confirmar que toda a arquitetura estava conectada adequadamente, foram iniciados os testes para investigar a qualidade da conexão e realizar comparações. A ideia dos primeiros testes foi utilizar o Aether como servidor e

realizar testes em que o UE-NON3GPP e o UE-gNB se conectassem ao Aether. Como demonstraremos no trecho de Codgio 4.3, vamos realizar um teste de conectividade entre o UE-NON3GPP e o Aether ao enviar um pacote “ping” para o núcleo da rede 5G.

```
1 ubuntu@UE-NON3GPP:~$ ping 192.168.250.1
2 PING 192.168.250.1 (192.168.250.1) 56(84) bytes of data.
3 64 bytes from 192.168.250.1: icmp_seq=1 ttl=64 time=1.50 ms
4 64 bytes from 192.168.250.1: icmp_seq=2 ttl=64 time=1.43 ms
5 64 bytes from 192.168.250.1: icmp_seq=3 ttl=64 time=1.72 ms
6 64 bytes from 192.168.250.1: icmp_seq=4 ttl=64 time=1.86 ms
7 64 bytes from 192.168.250.1: icmp_seq=5 ttl=64 time=1.62 ms
8 64 bytes from 192.168.250.1: icmp_seq=6 ttl=64 time=1.63 ms
9 64 bytes from 192.168.250.1: icmp_seq=7 ttl=64 time=1.48 ms
10 64 bytes from 192.168.250.1: icmp_seq=8 ttl=64 time=2.35 ms
11 64 bytes from 192.168.250.1: icmp_seq=9 ttl=64 time=1.44 ms
12 64 bytes from 192.168.250.1: icmp_seq=10 ttl=64 time=1.56 ms
13 64 bytes from 192.168.250.1: icmp_seq=11 ttl=64 time=1.86 ms
14 ^C
15 --- 192.168.250.1 ping statistics ---
16 11 packets transmitted, 11 received, 0% packet loss, time
    10013ms
17 rtt min/avg/max/mdev = 1.429/1.678/2.345/0.255 ms
```

4.3 – Ping do UE-NON3GPP para o Aether

É importante ressaltar que esses resultados são específicos para o teste de “ping” realizado no IP 192.168.250.1 e representam apenas uma parte da análise de desempenho e qualidade da conexão realizada neste trabalho. Outros testes e medições foram feitos para uma avaliação completa das conexões 3GPP eNON3GPP e seus respectivos desempenhos em relação ao núcleo do 5G. Essas informações serão apresentadas e analisadas nas seções subsequentes do trabalho.

Os resultados obtidos ao realizar o teste de ping no IP 192.168.250.1, que representa o núcleo do 5G como pode ser visto na Figura 18, mostraram uma conexão bem sucedida com o núcleo da rede. A análise estatística dos pacotes transmitidos e recebidos revelou uma taxa de sucesso de 100%, indicando que não houve perda de pacotes durante o teste.

Durante a execução deste comando de “ping”, foi utilizado o “tcpdump” para capturar os pacotes ICMP enviados e recebidos. Essa análise dos pacotes capturados pelo “tcpdump” permite verificar o funcionamento adequado da conexão entre o UE-NON3GPP e o núcleo do 5G, demonstrando que o UE está acessando a internet por meio do núcleo.

```
1 ubuntu@UE-NON3GPP:~$ sudo tcpdump -n -i any icmp
2 tcpdump: verbose output suppressed, use -v or -vv for full
  protocol decode
3 listening on any, link-type LINUX_SLL (Linux cooked v1),
  capture size 262144 bytes
4
5 21:18:56.489468 IP 172.250.237.124 > 192.168.250.1: ICMP echo
  request, id 142, seq 1, length 64
6 21:18:56.490946 IP 192.168.250.1 > 172.250.237.124: ICMP echo
  reply, id 142, seq 1, length 64
7 21:18:57.490524 IP 172.250.237.124 > 192.168.250.1: ICMP echo
  request, id 142, seq 2, length 64
8 21:18:57.491927 IP 192.168.250.1 > 172.250.237.124: ICMP echo
  reply, id 142, seq 2, length 64
9 21:18:58.492321 IP 172.250.237.124 > 192.168.250.1: ICMP echo
  request, id 142, seq 3, length 64
10 21:18:58.494015 IP 192.168.250.1 > 172.250.237.124: ICMP echo
  reply, id 142, seq 3, length 64
```

4.4 – Tcpcdump dentro do UE-NON3GPP

Também foi utilizado o “tcpdump” no Aether para capturar os pacotes ICMP trocados entre o núcleo do 5G e o UE-NON3GPP.

```
1 ubuntu@aether:~$ sudo tcpdump -n -i any icmp
2 tcpdump: verbose output suppressed, use -v or -vv for full
  protocol decode
3 listening on any, link-type LINUX_SLL (Linux cooked), capture
  size 262144 bytes
4
5 21:18:56.486412 IP 172.250.237.124 > 192.168.250.1: ICMP echo
  request, id 142, seq 1, length 64
6 21:18:56.486412 IP 172.250.237.124 > 192.168.250.1: ICMP echo
  request, id 142, seq 1, length 64
7 21:18:56.486462 IP 192.168.250.1 > 172.250.237.124: ICMP echo
  reply, id 142, seq 1, length 64
8 21:18:56.486465 IP 192.168.250.1 > 172.250.237.124: ICMP echo
  reply, id 142, seq 1, length 64
```

4.5 – Tcpcdump dentro do Aether

A execução do “tcpdump” no Aether nos permite monitorar a comunicação entre o núcleo do 5G e o UE-NON3GPP, onde se torna possível observar a captura de pacotes ICMP. De forma semelhante ao “tcpdump” realizado no UE-NON3GPP, cada linha da saída representa um pacote capturado, contendo detalhes importantes, como o endereço de origem, o endereço de destino, o tipo de pacote (seja este ICMP *echo request* ou ICMP *echo reply*), entre outras informações relevantes.

4.1 DADOS OBTIDOS

Os dados adquiridos serão apresentados em grupos distintos, associados aos componentes conectados de cada ambiente. A sequência em que os componentes são nomeados indica a direção da conexão estabelecida. Para ilustrar, quando mencionamos a conexão entre UE-NON3GPP e Aether, estamos nos referindo aos testes realizados a partir do UE-NON3GPP em direção ao Aether.

4.1.1 UE-NON3GPP para Aether

Primeiro, o comando “netperf -H 10.0.0.1 -l 60 -t TCP_STREAM” foi executado. Esse comando executa um teste de transmissão TCP, que mede a taxa de transferência entre os *hosts*.

```

1 ubuntu@UE-NON3GPP:~$ netperf -H 10.0.0.1 -l 60 -t TCP_STREAM
2 MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0
   AF_INET to 10.0.0.1 () port 0 AF_INET : demo
3 Recv      Send      Send
4 Socket  Socket  Message  Elapsed
5 Size     Size     Size     Time     Throughput
6 bytes   bytes   bytes    secs.    10^6bits/sec
7
8 131072   16384   16384    60.25    24.31

```

4.6 – TCP STREAM dentro do UE-NON3GPP

Neste caso, a taxa de transferência foi de 24.31 Mbits/seg, que foi medida durante um período de 60.25 segundos. Essa taxa de transferência é um indicador direto da largura de banda disponível entre os dois hosts.

Em seguida, foi realizado outro teste usando o comando “netperf -H 10.0.0.1 -l 60 -t TCP_RR”. Este comando executa um teste de requisição/resposta TCP (TCP_RR), que mede o número de transações TCP (uma requisição seguida por uma resposta) que podem ser realizadas por segundo.

```

1 ubuntu@UE-NON3GPP:~$ netperf -H 10.0.0.1 -l 60 -t TCP_RR

```

```

2 MIGRATED TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0)
   port 0 AF_INET to 10.0.0.1 () port 0 AF_INET : demo :
   first burst 0
3 Local /Remote
4 Socket Size Request Resp. Elapsed Trans.
5 Send Recv Size Size Time Rate
6 bytes Bytes bytes bytes secs. per sec
7
8 16384 131072 1 1 60.00 585.85
9 16384 131072

```

4.7 – TCP RR dentro do UE-NON3GPP

O resultado encontrado foi de 585.85 transações por segundo, por ultimo foi executado o seguinte comando:

```

1 sudo ping 192.168.250.1 -c 5000 -i 0.01
2
3 64 bytes from 192.168.250.1: icmp_seq=4996 ttl=64 time=1.52
   ms
4 64 bytes from 192.168.250.1: icmp_seq=4997 ttl=64 time=1.54
   ms
5 64 bytes from 192.168.250.1: icmp_seq=4998 ttl=64 time=1.48
   ms
6 64 bytes from 192.168.250.1: icmp_seq=4999 ttl=64 time=1.39
   ms
7 64 bytes from 192.168.250.1: icmp_seq=5000 ttl=64 time=2.79
   ms
8
9 --- 192.168.250.1 ping statistics ---
10 5000 packets transmitted, 5000 received, 0% packet loss, time
   50031ms
11 rtt min/avg/max/mdev = 1.079/2.255/15.013/1.210 ms, pipe 2

```

4.8 – Ping dentro do UE-NON3GPP

A partir do comando “ping” executado, pode-se perceber que não houve perda de pacotes. A latência média foi de 2.255 ms com um *jitter* de 1.210ms.

As estatísticas de uso de recursos da máquina também foram observadas. Foi registrado um uso de CPU de 82% de um consumo de memória RAM de 323M durante a execução dos testes na VM do UE-NON3GPP e da VM do Aether foi de 94% da CPU e 6.38GB de memória RAM.

4.1.2 UE-gNB para Aether

```

1 ubuntu@UE-gNB:~$ netperf -H 10.0.0.1 -L 172.250.237.122 -l 60
   -t TCP_STREAM
2 MIGRATED TCP STREAM TEST from 172.250.237.122 () port 0
   AF_INET to 10.0.0.1 () port 0 AF_INET : demo
3 ^CRecv Send Send
4 Socket Socket Message Elapsed
5 Size Size Size Time Throughput
6 bytes bytes bytes secs. 10^6bits/sec
7
8 131072 16384 16384 2.08 127.27

```

4.9 – TCP STREAM dentro do UE-gNB

No teste TCP_STREAM, a taxa de transferência média foi de 127.27 Mbps. Esta taxa de transferência sugere uma conexão bastante estável.

```

1 ubuntu@UE-gNB:~$ netperf -H 10.0.0.1 -L 172.250.237.123 -l 60
   -t TCP_RR
2 MIGRATED TCP REQUEST/RESPONSE TEST from 172.250.237.123 ()
   port 0 AF_INET to 10.0.0.1 () port 0 AF_INET : demo :
   first burst 0
3 Local /Remote
4 Socket Size Request Resp. Elapsed Trans.
5 Send Recv Size Size Time Rate
6 bytes Bytes bytes bytes secs. per sec
7
8 16384 131072 1 1 60.00 1617.56
9 16384 131072

```

4.10 – TCP RR dentro do UE-gNB

No teste “TCP_RR”, o sistema conseguiu realizar aproximadamente 1617.56 transações por segundo. Isto mostra uma alta capacidade para lidar com comunicações baseadas em requisição/resposta, que são críticas em muitas aplicações de rede.

```

1 sudo ping -I uesimtun0 192.168.250.1 -c 5000 -i 0.01
2 64 bytes from 192.168.250.1: icmp_seq=4997 ttl=64 time=0.815
   ms
3 64 bytes from 192.168.250.1: icmp_seq=4998 ttl=64 time=0.621
   ms

```

```

4 64 bytes from 192.168.250.1: icmp_seq=4999 ttl=64 time=0.857
   ms
5 64 bytes from 192.168.250.1: icmp_seq=5000 ttl=64 time=0.839
   ms
6
7 --- 192.168.250.1 ping statistics ---
8 5000 packets transmitted, 5000 received, 0% packet loss, time
   68289ms
9 rtt min/avg/max/mdev = 0.463/0.911/26.990/0.554 ms, pipe 2

```

4.11 – Ping dentro do UE-gNB

Com base no comando “ping” executado, foi possível constatar que não houve perda de pacotes e que a latência média foi de 0.911 ms com um desvio de 0.554 ms.

Além disso, o uso de recursos da máquina também foi registrado durante a execução dos testes. Foi observado na VM do UE-gNB um uso da CPU de 58% e um consumo de memória RAM de 240M e 77.5% da CPU e 5.78GB de RAM na VM do Aether. Essas métricas ajudam a avaliar como o sistema está lidando com as demandas da rede, com esses valores indicando um bom gerenciamento dos recursos disponíveis.

4.1.3 UE-gNB para UE-NON3GPP

```

1 ubuntu@UE-gNB:~$ netperf -H 172.250.237.124 -L
   172.250.237.123 -t TCP_STREAM -l 60
2 MIGRATED TCP STREAM TEST from 172.250.237.123 () port 0
   AF_INET to 172.250.237.124 () port 0 AF_INET : demo
3 Recv Send Send
4 Socket Socket Message Elapsed
5 Size Size Size Time Throughput
6 bytes bytes bytes secs. 10^6bits/sec
7
8 131072 16384 16384 60.21 17.34

```

4.12 – TCP STREAM dentro do UE-gNB

No teste “TC_STREAM” realizado do UE-gNB para UE-NON3GPP, a taxa de transferência média foi de 17.34 Mbps. O resultado representa uma conexão estável e uma taxa de transferência razoável.

```

1 ubuntu@UE-gNB:~$ netperf -H 172.250.237.124 -L
   172.250.237.123 -t TCP_RR -l 60

```

```

2 MIGRATED TCP REQUEST/RESPONSE TEST from 172.250.237.123 ()
   port 0 AF_INET to 172.250.237.124 () port 0 AF_INET : demo
   : first burst 0
3 Local /Remote
4 Socket Size Request Resp. Elapsed Trans.
5 Send Recv Size Size Time Rate
6 bytes Bytes bytes bytes secs. per sec
7
8 16384 131072 1 1 60.00 489.89
9 16384 131072

```

4.13 – TCP RR dentro do UE-gNB

Quanto ao teste “TCP_RR”, foram realizadas aproximadamente 489.89 transações por segundo.

```

1 sudo ping -I uesimtun0 172.250.237.136 -c 5000 -i 0.01
2 64 bytes from 172.250.237.136: icmp_seq=4991 ttl=63 time=2.22
   ms
3 64 bytes from 172.250.237.136: icmp_seq=4992 ttl=63 time=2.23
   ms
4 --- 172.250.237.136 ping statistics ---
5 4992 packets transmitted, 4992 received, +8 errors, 0% packet
   loss, time 49910ms
6 rtt min/avg/max/mdev = 1.726/2.773/15.389/0.934 ms, pipe 2

```

4.14 – Ping dentro do UE-gNB

Finalmente, a partir do comando “ping” executado, observa-se que não houve perda de pacotes, a latência média foi de 2.773 ms e o *jitter* médio foi de 0.934 ms. Estes resultados também apontam para uma rede altamente confiável e consistente.

Além disso, o uso de recursos da VM do UE-gNB durante os testes foi de 65% da CPU e 247M de memória RAM. Tais valores são importantes para avaliar a eficiência do sistema durante a execução dos testes.

4.1.4 UE-NON3GPP para UE-gNB

```

1 ubuntu@UE-NON3GPP:~$ netperf -H 172.250.237.123 -l 60 -t
   TCP_STREAM
2 MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0
   AF_INET to 172.250.237.123 () port 0 AF_INET : demo
3 Recv Send Send
4 Socket Socket Message Elapsed

```



```

5 Size Size Size Time Throughput
6 bytes bytes bytes secs. 10^6bits/sec
7
8 131072 16384 16384 60.06 19.83

```

4.15 – TCP STREAM dentro do UE-NON3GPP

No teste “TCP_STREAM” realizado do UE-NON3GPP para UE-gNB, a taxa de transferência média foi de 19.83 Mbps.

```

1 ubuntu@UE-NON3GPP:~$ netperf -H 172.250.237.123 -l 60 -t
   TCP_RR
2 MIGRATED TCP REQUEST/RESPONSE TEST from 0.0.0.0 (0.0.0.0)
   port 0 AF_INET to 172.250.237.123 () port 0 AF_INET : demo
   : first burst 0
3 Local /Remote
4 Socket Size Request Resp. Elapsed Trans.
5 Send Recv Size Size Time Rate
6 bytes Bytes bytes bytes secs. per sec
7
8 16384 131072 1 1 60.00 510.44
9 16384 131072

```

4.16 – TCP RR dentro do UE-NON3GPP

No teste “TCP_RR”, a rede conseguiu lidar com aproximadamente 510.44 transações por segundo.

```

1 sudo ping 172.250.237.137 -c 5000 -i 0.01
2 --- 172.250.237.137 ping statistics ---
3 4992 packets transmitted, 4992 received, +8 errors, 0% packet
   loss, time 49914ms
4 rtt min/avg/max/mdev = 1.753/2.670/10.482/0.825 ms, pipe 2

```

4.17 – Ping dentro do UE-NON3GPP

No comando “ping” executado, não houve perda de pacotes, e a latência média foi de 2.670 ms e 0.825 ms de *jitter*. Estes resultados indicam uma rede bastante estável e confiável. Durante a realização dos testes, o consumo de recursos da máquina foi de 78.66% da CPU e 321M de memória RAM na VM do UE-NON3GPP.

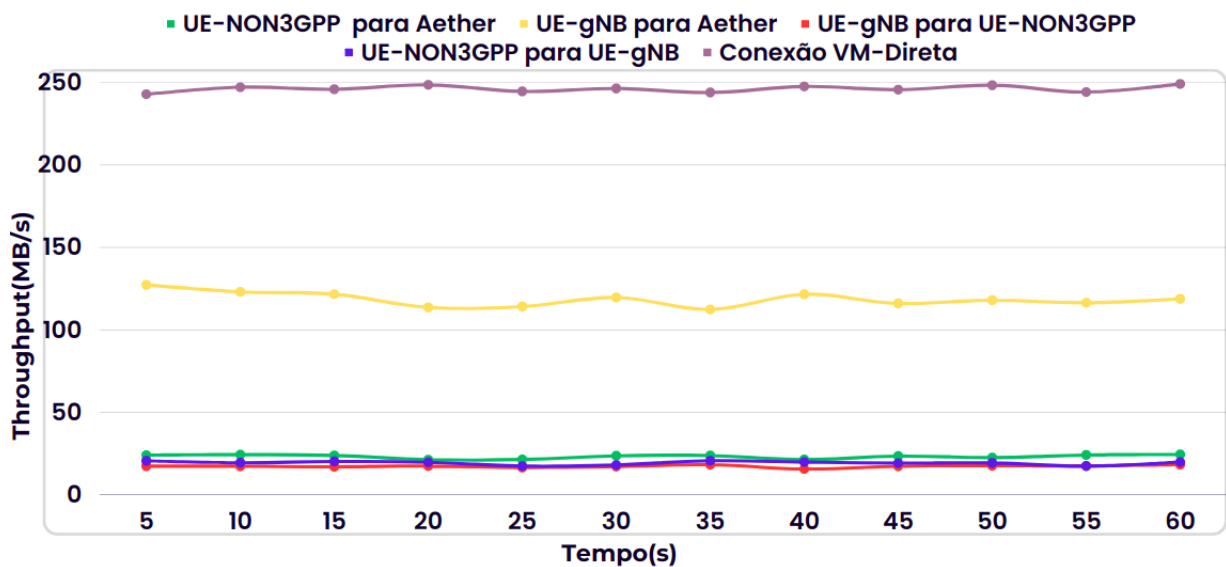
4.2 GRÁFICOS DOS DADOS OBTIDOS

Foram coletados e analisados uma série de parâmetros relevantes para a pesquisa. Visando facilitar a interpretação e a compreensão desses dados, os dados foram organizados

e apresentados em diversos gráficos e tabelas. Essa abordagem visual permite uma análise mais intuitiva e direta dos resultados obtidos. Além disso, também foram incluídos os valores obtidos na conexão direta entre as VMs sem a necessidade de passar pelo *núcleo* do 5G.

A seguir na Figura 19 é possível ver o gráfico de das taxas de transferência obtidos em intervalos de 5 segundos.

Figura 19 – Gráfico de *throughput*.



Fonte: O autor.

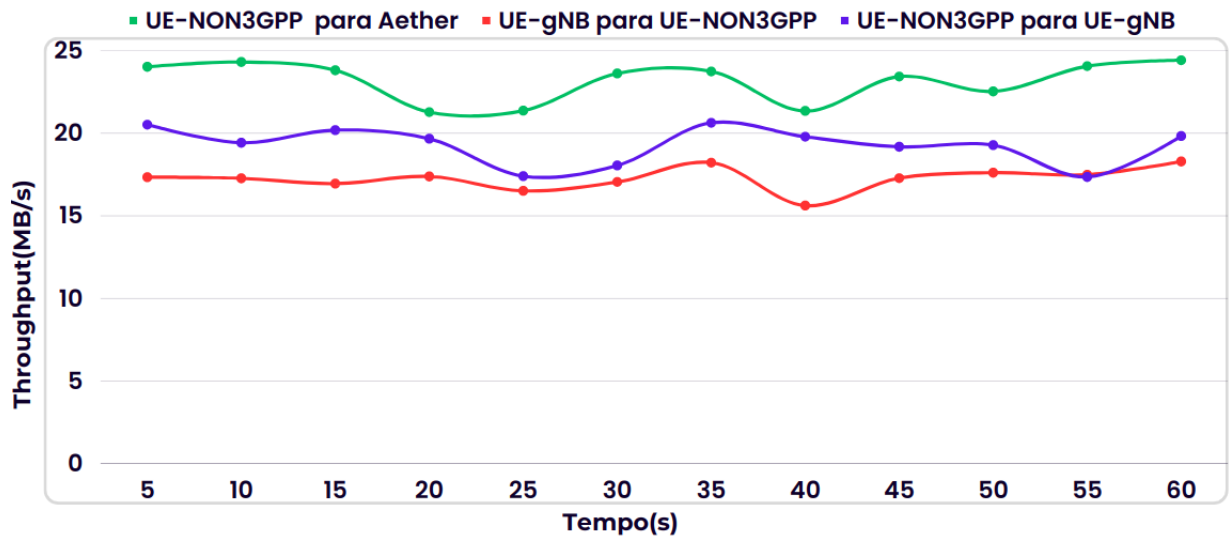
Para uma visualização mais clara, o gráfico apresentado na Figura 20 teve os dados correspondentes à taxa de transferência do UE-gNB para o Aether e da conexão VM-Direta removidos.

Nos gráficos de *throughput* previamente apresentados, é perceptível que a VM-Direta, utilizada como referência, apresentou o um melhor desempenho, conforme previsto. O segundo resultado mais expressivo pertenceu à conexão UE-gNB com o núcleo, seguido pela conexão entre o UE-NON3GPP e o núcleo. As conexões de UE para UE, por sua vez, exibiram os resultados mais modestos de *throughput*.

Utilizando o “ping” para medir a latência entre os componentes foi possível obter os dados apresentados pela Figura 21.

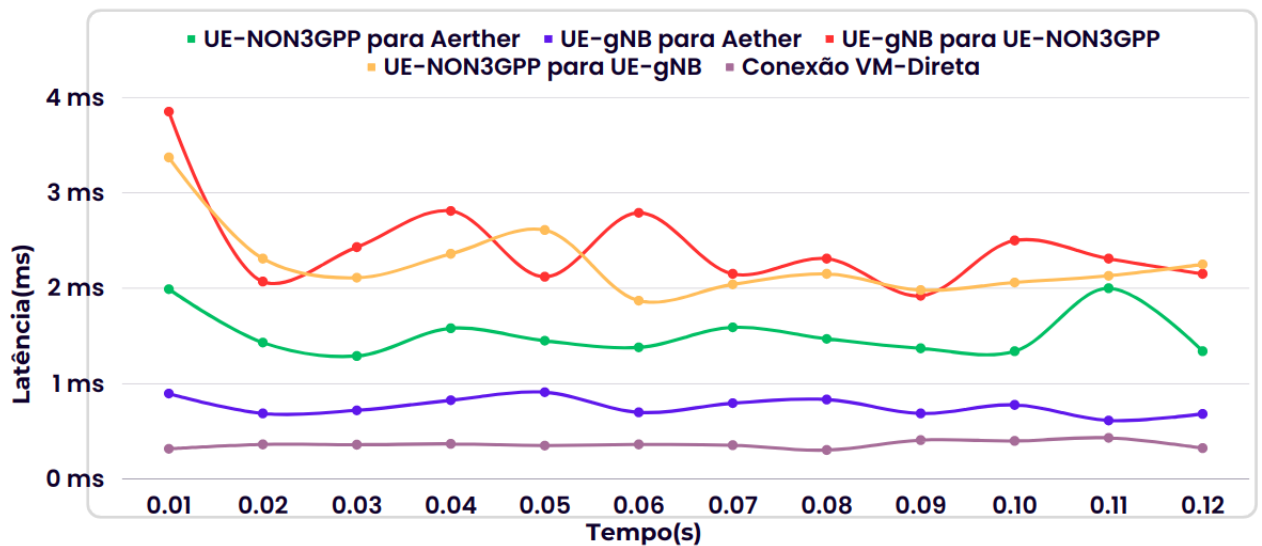
No que concerne ao gráfico de latência exibido anteriormente, é possível identificar que a VM-Direta, empregada como parâmetro de referência, manifestou o menor valor de latência, conforme esperado. O segundo menor valor foi observado para a conexão UE-gNB com o núcleo, seguida pela conexão entre o UE-NON3GPP e o núcleo. As conexões de UE para UE apresentaram os valores mais elevados de latência.

Figura 20 – Gráfico de *throughput* sem UE-gNB para o Aether e VM-Direta.



Fonte: O autor.

Figura 21 – Gráfico de latência.



Fonte: O autor.

Na Tabela 2 são apresentados os resultados médios coletados de cada teste realizado. Esta tabela fornece uma visão clara dos dados coletados e permite uma comparação fácil e intuitiva entre os diferentes tipos de conexão testados. Ao examinar esses valores, podemos discernir padrões e tendências que nos permitem entender melhor o desempenho relativo das conexões 3GPP e NON3GPP no contexto do 5G.

Analisando a tabela apresentada anteriormente, constata-se que a VM-Direta, uti-

Tabela 2 – Resultados obtidos. UN = UE-NON3GPP e Ug = UE-gNB.

	UN-Aether	UN-Ug	Ug-Aether	Ug-UN	VM-Direta
Conexão	OK	OK	OK	OK	OK
Banda	24.31 Mbps	19.83 Mbps	127.27 Mbps	17.34 Mbps	245.64 Mbps
Transações	585.85/s	510.44/s	1617.56/s	489.89/s	5221.64/s
Latência	2.255 ms	2.670 ms	0.911 ms	2.773 ms	0.371 ms
<i>Jitter</i>	1.210 ms	0.825 ms	0.554 ms	0.934 ms	0.136 ms
RAM	323 MB	321 MB	240 MB	247 MB	260 MB
CPU	82%	78.66%	58%	65%	16%

Fonte: O autor.

lizada como referencial, apresentou os indicadores mais favoráveis, conforme era esperado. O segundo conjunto de indicadores mais vantajosos foi observado na conexão UE-gNB com o núcleo, seguido por notável diferença pela conexão entre o UE-NON3GPP e o núcleo. Por último, as conexões de UE para UE registraram os indicadores menos favoráveis em todos os testes.

5 CONCLUSÃO

Neste estudo, todos os objetivos previamente estabelecidos foram atingidos com êxito. A simulação de uma conexão 3GPP e uma conexão NON3GPP foi realizada e demonstrou-se que ambas conseguem se integrar de maneira eficiente ao núcleo do 5G. Ademais, confirmou-se a viabilidade de uma comunicação eficaz e direta entre as entidades 3GPP e NON3GPP.

No que concerne aos dados de QoS obtidos, observou-se uma eficiência superior na conexão nativa do 5G (3GPP) quando comparada à conexão NON3GPP. Tal disparidade de desempenho pode ser atribuída ao *overhead* de configuração requerido para estabelecer a conexão NON3GPP, assim como o estabelecimento do túnel *NWu* por meio da conexão entre o UE e o N3IWF.

Foi ainda evidenciado um aumento no esforço computacional, tanto em termos de CPU quanto de memória RAM, para estabelecer a conexão NON3GPP. Essa constatação se fez presente tanto no núcleo 5G (Aether) quanto na UE-NON3GPP. Adicionalmente, a pesquisa revelou que a interconexão do NON3GPP é segura, uma vez que são estabelecidos túneis *Ipssec* e *IKEv2*.

Os achados deste estudo corroboram a viabilidade da integração de conexões NON3GPP nas redes 5G, mesmo diante dos desafios inerentes a tal empreitada. Tal fato adquire especial relevância na era da Internet das Coisas (IoT), na qual uma ampla gama de dispositivos, desde sensores industriais até dispositivos de consumo, precisam estabelecer conexões e comunicar-se de forma eficaz. A eficiência proposta pela rede 5G tem o potencial de fomentar significativamente a operação e agilizar a tomada de decisões em ambientes industriais.

Os resultados advindos deste trabalho contribuem para a expansão do entendimento acerca da infraestrutura do 5G, pavimentando o caminho para futuras investigações sobre a otimização das conexões 3GPP e NON3GPP. Estas podem abordar métodos para aprimorar ainda mais a conectividade e a eficiência da IoT, particularmente em contextos industriais, onde a baixa latência, alta confiabilidade e capacidade de gerenciar inúmeros dispositivos conectados são imperativos. Além disso, a possibilidade de integrar o ambiente industrial com a SD-WAN utilizando ambas as conexões 3GPP e NON3GPP proporciona uma oportunidade para maximizar o desempenho, versatilidade e confiabilidade da rede.

Os resultados aqui obtidos sugerem um progresso contínuo no desenvolvimento e aprimoramento dos serviços de telecomunicações baseados na tecnologia 5G, favorecendo a implementação de aplicações de IoT industriais robustas e eficientes.

REFERÊNCIAS

AETHER Documentation. Acessado em: 21 de Junho de 2023. 2023. Disponível em: <https://docs.aetherproject.org/master/index.html>.

AETHER PROJECT. **Aether in a Box Hardware Requirements**. [*S.l.: s.n.*], Acesso em: 23 jun. 2023. Disponível em: <https://docs.aetherproject.org/master/developer/aiabhw5g.html>.

BOTH, C. ET AL. **Soft5G+: explorando a softwarização nas redes 5G**. [*S.l.*]: XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2020. P. 27, 28, 29, 30.

CHANDRAMOULI, D.; LIEBHART, R.; PIRSKANEN, J. **5G for the Connected World**. [*S.l.*]: John Wiley Sons Ltd, 2019. P. 32, 34, 37, 38.

CONTAINER. Acessado em: 21 de Junho de 2023. 2023. Disponível em: <https://www.docker.com/resources/what-container/>.

COX, C. **An Introduction to 5G: The New Radio, 5G Network and Beyond**. [*S.l.*]: Wiley, 2020. P. 19, 37. ISBN 9781119602682.

DAVANÇO SILVA, G. H. **Classificação de tráfego por classes de serviço no núcleo 5G**. [*S.l.: s.n.*], 2021. Disponível em: <https://repositorio.ufsc.br/bitstream/handle/123456789/237577/TCC.pdf>.

FERNANDO AUGUSTO SILVA FERRAZ. **NWu-Non3GPP-5GC**. [*S.l.: s.n.*], Acesso em: 23 jun. 2023. Disponível em: <https://github.com/fasferraz/NWu-Non3GPP-5GC>.

FREE5GC. **Diff for Pull Request 3 on Free5GC AUSF**. [*S.l.: s.n.*], Acesso em: 23 jun. 2023. Disponível em: <https://diff.githubusercontent.com/raw/free5gc/ausf/pull/3.diff>.

FREE5GC. **Diff for Pull Request 35 on Free5GC AMF**. [*S.l.: s.n.*], Acesso em: 23 jun. 2023. Disponível em: <https://diff.githubusercontent.com/raw/free5gc/amf/pull/35.diff>.

FREE5GC. **Free5GC - The Open Source 5G Core Network**. [*S.l.: s.n.*], Acesso em: 23 jun. 2023. Disponível em: <https://free5gc.org/>.

FREE5GC. **Free5GC N3IWF**. [S.l.: s.n.], Acesso em: 23 jun. 2023. Disponível em:<https://github.com/free5gc/n3iwf>.

KRAUS, D. **Computação de borda para indústria utilizando a rede 5G**. [S.l.: s.n.], 2021. Disponível em:<https://repositorio.ufsc.br/handle/123456789/228613>.

KUBERNETES. Acessado em: 21 de Junho de 2023. 2023. Disponível em: <https://kubernetes.io/pt-br/>.

MAILER, C. **Desenvolvimento e análise de um modelo de simulação para arquiteturas 5G industriais**. [S.l.: s.n.], 2022. P. 26. Disponível em:<https://repositorio.ufsc.br/handle/123456789/244587>.

MAILER, C. **Plataforma de CORE 5G em nuvem para disponibilização de funções de rede como serviço**. [S.l.: s.n.], 2020. P. 21, 22, 23, 24, 25, 30. Disponível em:https://repositorio.ufsc.br/bitstream/handle/123456789/209624/TCC_20201_ChristianMailer.pdf.

MAZIERO BARETTA, J. L. **Encapsulamento de protocolos Ethernet Industrial utilizando a rede 5G**. [S.l.: s.n.], 2022. Disponível em:<https://repositorio.ufsc.br/handle/123456789/243552>.

NUNES, Giovanni Reis. **Virtualbox na linha de comando**. Acessado em: 21 de Junho de 2023. 2015. Disponível em: <https://giovannireisnunes.wordpress.com/2015/10/30/virtualbox-na-linha-de-comando/>.

PENTTINEN, Jyrki T. J. **5G Explained: Security and Deployment of Advanced Mobile Communications**. [S.l.]: John Wiley & Sons Ltd, 2019. P. 32, 48, 49, 50, 51, 52, 83, 84, 85, 127, 128, 154, 155, 156, 157, 158, 159, 160. ISBN 9781119275688.

ROMMER, S. et al. **5G Core Networks: Powering Digitalization**. London: Academic Press, 2020. P. 22, 23, 40, 41, 42, 43, 44, 45, 46, 47, 52, 53, 54, 66, 67, 68. ISBN 978-0081030097.

SEGEČ, P.; MORAVČIK, M.; URAMOVÁ, J.; PAPÁN, J.; YEREMENKO, O. SD-WAN – architecture, functions and benefits. **IEEE Xplore**, 2021.

UBUNTU. **Ubuntu 18.04 LTS (Bionic Beaver)**. [S.l.: s.n.], Acesso em: 23 jun. 2023. Disponível em:<https://releases.ubuntu.com/18.04/>.

UERANSIM. **UERANSIM**. Acessado em: 21 de Junho de 2023. 2023. Disponível em:
<https://github.com/aligungr/UERANSIM>.

VIRTUALBOX. Acessado em: 21 de Junho de 2023. 2023. Disponível em:
<https://www.virtualbox.org/>.

APÊNDICE A – Configuração do Aether

```
1     # SPDX-FileCopyrightText: 2022-present Intel Corporation
2 # Copyright 2019-present Open Networking Foundation
3 #
4 # SPDX-License-Identifier: Apache-2.0
5
6 omecc-control-plane:
7   enable4G: false
8
9 # Override values for 5g-control-plane helm chart
10 # 5g-control-plane Helm Charts values - https://gerrit.
    opencord.org/sdcore-helm-charts/5g-control-plane/values.
    yaml
11
12 5g-control-plane:
13   enable5G: true
14   images:
15     repository: "registry.opennetworking.org/docker.io/"
16     #tags:
17     # amf: amf:latest
18     # ausf: ausf:latest
19
20 kafka:
21   deploy: true
22
23 mongodb:
24   usePassword: false
25   persistence:
26     enabled: false
27   #enable following 2 lines if cluster needs to be enabled
28   architecture: replicaset
29   replicaCount: 2
30 resources:
31   enabled: false
32
33 config:
34   mongodb:
35     name: free5gc
```

```
36     url: mongodb://mongodb-arbiter-headless #enable if
cluster deployment required
37
38     managedByConfigPod:
39         enabled: true
40         # syncUrl: http://sdcore-adapter-v2-1.aether-roc.svc
:8080/synchronize
41
42     sctplb:
43         deploy: false # if enabled then deploy sctp pod. Once
backend code is stable,
44             # this option will be enabled by default
45
46     upfadapter:
47         deploy: false # if enabled then deploy upf adapter pod.
Once backend code is stable,
48             # this option will be enabled by default
49
50     metricfunc:
51         deploy: true
52
53     # Change AMF config here if rquired Most of the AMF
config comes from Slice APIs
54     # but some of the config is directly provided thorough
helm chart
55     amf:
56         # use externalIP if you need to access your AMF from
remote setup and you don't
57         # want setup NodePort Service Type
58     ngapp:
59         externalIp: "10.15.215.27"
60         port: 38412
61     cfgFiles:
62         amfcfg.conf:
63             configuration:
64                 # storing AMF subscribers in the datastore is
disabled by default. Once internal
65                 # testing is complete we shall enable DBStore by
default.
```

```
66         enableDBStore: false
67
68     # SMF config override. Refer helm chart values for more
options
69     smf:
70         cfgFiles:
71             smfcfg.conf:
72                 configuration:
73                     # storing AMF subscribers in the datastore is
disabled by default. Once internal
74                     # testing is complete we shall enable DBStore by
default.
75                     enableDBStore: false
76
77     #pcf:
78     #   cfgFiles:
79     #     pcfcfg.conf:
80     #       configuration:
81
82     nrf:
83         cfgFiles:
84             nrfcfg.conf:
85                 configuration:
86                     mongoDBStreamEnable: false # enable/disable
MongoDB stream in NRF. Keep it disabled for now.
87                     nfProfileExpiryEnable: false # if enabled then
remove NF profile if no keepalive received
88                     nfKeepAliveTime: 60 #default timeout for NF
profiles. This value will be sent to registered NF during
registration/update process.
89
90
91 # Override values for omecc-sub-provision helm chart
92 # omecc-control-plane Helm Charts values - https://gerrit.opencord.org/sdcore-helm-charts/omecc-sub-provision/values.yaml
93 omecc-sub-provision:
94     enable: true
95     images:
```

```
96     repository: "registry.opennetworking.org/docker.io/"
97     #tags:
98     #   simapp: #add simapp override image
99
100  config:
101    simapp:
102      cfgFiles:
103        simapp.yaml:
104          configuration:
105            provision-network-slice: true # if enabled then
106            Network Slices configured from simapp
107            sub-provision-endpt:
108              addr: webui.omec.svc.cluster.local #
109            subscriber configuration endpoint.
110            # sub-proxy-endpt: #this is useful block if
111            subscriber proxy is enabled in the ROC.
112            #   addr: subscriber-proxy.aether-roc.svc.cluster
113            .local
114            #   port: 5000
115
116            # below block configures the subscribers and
117            their security details.
118            # you can have any number of subscriber ranges
119            subscribers:
120            - ueId-start: "315010999912301"
121              ueId-end: "315010999912303"
122              plmnId: "315010"
123              opc: "69d5c2eb2e2e624750541d3bbc692ba5"
124              op: ""
125              key: "000102030405060708090a0b0c0d0e0f"
126              sequenceNumber: "135"
127            - ueId-start: "208930100007501"
128              ueId-end: "208930100007599"
129              plmnId: "20893"
130              opc: "981d464c7c52eb6e5036234984ad0bcf"
131              op: ""
132              key: "5122250214c33e723a5dd523fc145fc0"
133              sequenceNumber: "16f3b3f70fc2"
```

```
130         # Configure More than one device Group as per
        your need
131         device-groups:
132         - name: "5g-gnbsim-user-group1"
133           imsis:
134             - "208930100007487"
135             - "208930100007488"
136             - "208930100007489"
137             - "208930100007490"
138             - "208930100007491"
139             - "208930100007492"
140             - "208930100007493"
141             - "208930100007494"
142             - "208930100007495"
143             - "208930100007496"
144             - "208930100007497"
145             - "208930100007498"
146             - "208930100007499"
147             - "208930100007500"
148           ip-domain-name: "pool1"
149           ip-domain-expanded:
150             dnn: internet
151             dns-primary: "8.8.8.8" # Change this as per
        your need. This value is sent to UE
152             mtu: 1460 # This value is sent to UE/Mobile/
        Subscriber during Subscriber PDU Session Est
153             ue-ip-pool: "172.250.1.0/16" # IP address
        pool for subscribers
154           ue-dnn-qos:
155             dnn-mbr-downlink: 200000000 #ue level
        downlink qos
156             dnn-mbr-uplink: 200000000 #ue level
        uplink qos
157             bitrate-unit: bps #unit for qos rates
        mentioned above
158             traffic-class: #default bearer QCI/ARP.
        These values are not used in 5G as such
159             name: "platinum"
160             qci: 9
```

```
161         arp: 6
162         pdb: 300
163         pelr: 6
164         site-info: "aiab"
165         #configure only 1 device group if you want UPF to
allocate UE address.
166         #if smf is allocating IP address then having 2
separate pools is not a problem.
167         - name: "5g-gnbsim-user-group2" # second device
Group
168         imsis:
169             - "208930100007501"
170             - "208930100007502"
171             - "208930100007503"
172             - "208930100007504"
173             - "208930100007505"
174             - "208930100007506"
175             - "208930100007507"
176             - "208930100007508"
177             - "208930100007509"
178             - "208930100007510"
179         ip-domain-name: "pool2"
180         ip-domain-expanded:
181             dnn: internet
182             dns-primary: "8.8.8.8" # Change this as per
your need. This value is sent to UE
183             mtu: 1460 # This value is sent to UE/Mobile/
Subscriber during Subscriber PDU Session Est
184             ue-ip-pool: "172.250.1.0/16" # IP address
pool for subscribers
185         ue-dnn-qos:
186             dnn-mbr-downlink: 400000000
187             dnn-mbr-uplink: 100000000
188             bitrate-unit: bps # this value can be in
gbps, mbps, kbps
189             traffic-class: #default bearer QCI/ARP
190                 name: "platinum"
191                 qci: 8
192                 arp: 6
```

```
193         pdb: 300
194         pelr: 6
195         site-info: "aiab2"
196         network-slices:
197         - name: "default" #provide any unique slice name
198           slice-id: #make sure this slice matches with
199             slice configured in gNB, UE
200           sd: "010203"
201           sst: 1
202           site-device-group:
203           - "5g-gnbsim-user-group1" #all UEs mentioned
204             in this device group are assigned to this slice
205           - "5g-gnbsim-user-group2"
206             # below list of applicaiton filter controls
207             what user can access.
208             # by default user can access all networks.
209             application-filtering-rules:
210             #all application information. Default, allow
211             all applications
212             - rule-name: "ALLOW-ALL"
213               priority: 250
214               action: "permit"
215               endpoint: "0.0.0.0/0"
216             site-info:
217             # Provide gNBs and UPF details and also PLMN
218             for the site
219             gNodeBs:
220             - name: "aiab-gnb1"
221               tac: 1
222             - name: "aiab-gnb2"
223               tac: 2
224             plmn:
225             mcc: "208"
226             mnc: "93"
227             site-name: "aiab"
228             upf:
229             upf-name: "upf" # associated UPF for this
230             slice. One UPF per Slice. Provide fully qualified name
231             upf-port: 8805
```

```
226
227 # Override values for omeo-user-plane helm chart
228 # omeo-user-plane Helm Charts values - https://gerrit.
    opencord.org/sdfabric-helm-charts/bess-upf/values.yaml
229
230 omeo-user-plane:
231   enable: true
232   resources:
233     enabled: false
234   images:
235     repository: "registry.opennetworking.org/docker.io/"
236     # uncomment below section to add update bess image tag
237     #tags:
238     #  bess: <bess image tag>
239     #  pfcpiface: <pfcp image tag>
240   config:
241     upf:
242       name: "oaisim"
243       sriov:
244         enabled: false #default sriov is disabled in AIAB
245     setup
246       hugepage:
247         enabled: false #should be enabled if dpdk is enabled
248         #can be any other plugin as well, remember this plugin
249         dictates how IP address are assigned.
250       cniPlugin: macvlan
251       ipam: static
252       routes:
253         - to: ${NODE_IP}
254           via: 169.254.1.1
255       enb:
256         subnet: ${RAN_SUBNET} #this is your gNB network
257       access:
258         iface: ${DATA_IFACE}
259         # ip: 192.168.252.3/24 # this is the default in the
Helm chart
258   core:
259     iface: ${DATA_IFACE}
```



```
260     # ip: 192.168.250.3/24   # this is the default in the
    Helm chart
261     cfgFiles:
262     upf.json:
263         mode: af_packet   #this mode means no dpdk
264         hwcksum: true
265         log_level: "trace"
266         gtppsc: true #extension header is enabled in 5G.
    Sending QFI in pdu session extension header
267     cpiface:
268         dnn: "internet" #keep it matching with Slice dnn
269         hostname: "upf"
270         #http_port: "8080"
271         enable_ue_ip_alloc: false # if true then it means
    UPF allocates address from below pool
272         ue_ip_pool: "172.250.0.0/16" # UE ip pool is used
    if enable_ue_ip_alloc is set to true
273
274 # Override values for 5g-ran-sim helm chart
275 # 5g-ran-sim Helm Charts values - https://gerrit.opencord.org
    /sdcore-helm-charts/5g-ran-sim/values.yaml
276
277 5g-ran-sim:
278     enable: ${ENABLE_GNBSIM}
279     resources:
280         enabled: false
281     images:
282         repository: "registry.opennetworking.org/docker.io/"
283         #tags:
284         # gnbsim: 5gc-gnbsim:0.0.1-dev-local5
285     config:
286         gnbsim:
287             networkTopo:
288                 - upfAddr: "192.168.252.0/24"
289                 upfGw: "192.168.251.1"
290             singleInterface: false #default multiInterface. Works
    well for AIAB
291             execInParallel: false #run all profiles in parallel
292             goProfile:
```

```
293     enable: false #enable/disable golang profile in
gnbsim
294     port: 5000
295     httpServer:
296     enable: false #enable httpServer in gnbsim
297     port: 6000
298     yamlCfgFiles:
299     gnb.conf:
300     logger:
301     logLevel: trace # how detailed the log will be,
values: trace, debug, info, warn, error, fatal, panic
302     configuration:
303     #gnbs: # pool of gNodeBs
304     # gnb1:
305     #   n3IpAddr: "POD_IP" # set if singleInterface
is true
306     gnbs: # pool of gNodeBs
307     gnb1:
308     defaultAmf:
309     hostname: amf # Host name of AMF
310     port: 38412 # AMF port
311     customProfiles:
312     customProfiles1:
313     profileType: custom # profile type
314     profileName: custom1 # uniquely identifies
a profile within application
315     enable: false # Set true to execute the
profile, false otherwise.
316     execInParallel: false #run all subscribers
in parallel.
317     stepTrigger: false #wait for trigger to
move to next step
318     gnbName: gnb1 # gNB to be used for this
profile
319     startImsi: 208930100007487
320     ueCount: 30
321     defaultAs: "192.168.250.1" #default icmp
pkt destination
322     opc: "981d464c7c52eb6e5036234984ad0bcf"
```

```
323         key: "5122250214c33e723a5dd523fc145fc0"
324         sequenceNumber: "16f3b3f70fc2"
325         plmnId: # Public Land Mobile Network ID, <
PLMN ID> = <MCC><MNC>
326         mcc: 208 # Mobile Country Code (3 digits
string, digit: 0~9)
327         mnc: 93 # Mobile Network Code (2 or 3
digits string, digit: 0~9)
328         # start from "startiteration" and then move
through all the iteration till quit
329         startiteration: iteration1
330         iterations:
331         - "name": "iteration1"
332           "1": "REGISTRATION-PROCEDURE 5"
333           "2": "PDU-SESSION-ESTABLISHMENT-
PROCEDURE 5"
334           "3": "USER-DATA-PACKET-GENERATION-
PROCEDURE 10"
335         "next": "iteration2"
336         - "name": "iteration2"
337           "1": "AN-RELEASE-PROCEDURE 10"
338           "2": "UE-TRIGGERED-SERVICE-REQUEST-
PROCEDURE 5"
339         "repeat": 0 #default value 0 . i.e
execute once
340         "next": "iteration3" #default value
quit. i.e. no further iteration to run
341         - "name": "iteration3"
342           "1": "UE-INITIATED-DEREGISTRATION-
PROCEDURE 10"
343         "repeat": 0 #default value 0 . i.e
execute once
344         "next": "quit" #default value quit. i.
e. no further iteration to run
345         profiles: # profile information
346         - profileType: register # profile type
347         profileName: profile1 # uniquely identifies a
profile within application
```

```
348         enable: false # Set true to execute the
profile, false otherwise.
349         gnbName: gnb1 # gNB to be used for this
profile
350         execInParallel: false #run all subscribers in
this profile parallel
351         startImsi: 208930100007487 #First IMSI.
Subsequent values will be used if ueCount is mo re than
1
352         ueCount: 5 # Number of UEs for which the
profile will be executed
353         defaultAs: "192.168.250.1" #default icmp pkt
destination
354         perUserTimeout: 100 #if no expected event
received in this time then treat it as failure
355         plmnId: # Public Land Mobile Network ID, <
PLMN ID> = <MCC><MNC>
356             mcc: 208 # Mobile Country Code (3 digits
string, digit: 0~9)
357             mnc: 93 # Mobile Network Code (2 or 3
digits string, digit: 0~9)
358             opc: "981d464c7c52eb6e5036234984ad0bcf"
359             key: "5122250214c33e723a5dd523fc145fc0"
360             sequenceNumber: "16f3b3f70fc2"
361     - profileType: pdusesest
362       profileName: profile2
363       enable: true
364       gnbName: gnb1
365       execInParallel: false #run all subscribers in
this profile parallel
366       startImsi: 208930100007487
367       ueCount: 5
368       defaultAs: "192.168.250.1" #default icmp pkt
destination
369       perUserTimeout: 100 #if no expected event
received in this time then treat it as failure
370       plmnId:
371         mcc: 208
372         mnc: 93
```

```
373         dataPktCount: 5 # Number of UL user data
        packets to be transmitted. Common for all UEs
374         opc: "981d464c7c52eb6e5036234984ad0bcf"
375         key: "5122250214c33e723a5dd523fc145fc0"
376         sequenceNumber: "16f3b3f70fc2"
377     - profileType: anrelease
378       profileName: profile3
379       enable: false
380       execInParallel: false #run all subscribers in
        this profile parallel
381       gnbName: gnb1
382       startImsi: 208930100007497
383       ueCount: 5
384       defaultAs: "192.168.250.1" #default icmp pkt
        destination
385       perUserTimeout: 100 #if no expected event
        received in this time then treat it as failure
386       plmnId:
387         mcc: 208
388         mnc: 93
389       opc: "981d464c7c52eb6e5036234984ad0bcf"
390       key: "5122250214c33e723a5dd523fc145fc0"
391       sequenceNumber: "16f3b3f70fc2"
392     - profileType: uetriggservicereq
393       profileName: profile4
394       enable: false
395       gnbName: gnb1
396       execInParallel: false #run all subscribers in
        this profile parallel
397       startImsi: 208930100007497
398       ueCount: 5
399       plmnId:
400         mcc: 208
401         mnc: 93
402       defaultAs: "192.168.250.1" #default icmp pkt
        destination
403       perUserTimeout: 100 #if no expected event
        received in this time then treat it as failure
404       opc: "981d464c7c52eb6e5036234984ad0bcf"
```

```
405         key: "5122250214c33e723a5dd523fc145fc0"
406         sequenceNumber: "16f3b3f70fc2"
407     - profileType: deregister
408       profileName: profile5
409       enable: false
410       gnbName: gnb1
411       execInParallel: false #run all subscribers in
    this profile parallel
412       startImsi: 208930100007497
413       ueCount: 5
414       defaultAs: "192.168.250.1" #default icmp pkt
    destination
415       perUserTimeout: 100 #if no expected event
    received in this time then treat it as failure
416       plmnId:
417         mcc: 208
418         mnc: 93
419         opc: "981d464c7c52eb6e5036234984ad0bcf"
420       key: "5122250214c33e723a5dd523fc145fc0"
421       sequenceNumber: "16f3b3f70fc2"
422     - profileType: nwtriggeruedereg # profile type
423       profileName: profile6 # uniquely identifies a
    profile within application
424       enable: false # Set true to execute the
    profile, false otherwise.
425       gnbName: gnb1 # gNB to be used for this
    profile
426       execInParallel: false #run all subscribers in
    this profile parallel
427       startImsi: 208930100007497 # First IMSI.
    Subsequent values will be used if ueCount is more than 1
428       ueCount: 5 # Number of UEs for which the
    profile will be executed
429       defaultAs: "192.168.250.1" #default icmp pkt
    destination
430       perUserTimeout: 100 #if no expected event
    received in this time then treat it as failure
431       plmnId: # Public Land Mobile Network ID, <
    PLMN ID> = <MCC><MNC>. Should match startImsi
```

```
432         mcc: 208 # Mobile Country Code (3 digits
string, digit: 0~9)
433         mnc: 93 # Mobile Network Code (2 or 3
digits string, digit: 0~9)
434         opc: "981d464c7c52eb6e5036234984ad0bcf"
435         key: "5122250214c33e723a5dd523fc145fc0"
436         sequenceNumber: "16f3b3f70fc2"
437     - profileType: uereqpdusessrelease # profile
type
438         profileName: profile7 # uniquely identifies a
profile within application
439         enable: false # Set true to execute the
profile, false otherwise.
440         gnbName: gnb1 # gNB to be used for this
profile
441         execInParallel: false #run all subscribers in
this profile parallel
442         startImsi: 208930100007497 # First IMSI.
Subsequent values will be used if ueCount is more than 1
443         ueCount: 5 # Number of UEs for which the
profile will be executed
444         plmnId: # Public Land Mobile Network ID, <
PLMN ID> = <MCC><MNC>. Should match startImsi
445         mcc: 208 # Mobile Country Code (3 digits
string, digit: 0~9)
446         mnc: 93 # Mobile Network Code (2 or 3
digits string, digit: 0~9)
447         defaultAs: "192.168.250.1" #default icmp pkt
destination
448         perUserTimeout: 100 #if no expected event
received in this time then treat it as failure
449         opc: "981d464c7c52eb6e5036234984ad0bcf"
450         key: "5122250214c33e723a5dd523fc145fc0"
451         sequenceNumber: "16f3b3f70fc2"
```

APÊNDICE B – Configuração do UERANSIM

```
1 # IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15
   digits)
2 supi: 'imsi-208930100007501'
3 # Mobile Country Code value of HPLMN
4 mcc: '208'
5 # Mobile Network Code value of HPLMN (2 or 3 digits)
6 mnc: '93'
7 # Routing Indicator
8 routingIndicator: '0000'
9
10 # Permanent subscription key
11 key: '5122250214c33e723a5dd523fc145fc0'
12 # Operator code (OP or OPC) of the UE
13 op: '981d464c7c52eb6e5036234984ad0bcf'
14 # This value specifies the OP type and it can be either 'OP'
   or 'OPC'
15 opType: 'OPC'
16 # Authentication Management Field (AMF) value
17 amf: '8000'
18 # IMEI number of the device. It is used if no SUPI is
   provided
19 imei: '356938035643803'
20 # IMEISV number of the device. It is used if no SUPI and IMEI
   is provided
21 imeiSv: '4370816125816151'
22
23 # List of gNB IP addresses for Radio Link Simulation
24 gnbSearchList:
25   - 192.168.56.102
26
27 # UAC Access Identities Configuration
28 uacAic:
29   mps: false
30   mcs: false
31
32 # UAC Access Control Class
33 uacAcc:
```



```
34  normalClass: 0
35  class11: false
36  class12: false
37  class13: false
38  class14: false
39  class15: false
40
41 # Initial PDU sessions to be established
42 sessions:
43   - type: 'IPv4'
44     apn: 'internet'
45     slice:
46       sst: 0x01
47       sd: 0x010203
48
49 # Configured NSSAI for this UE by HPLMN
50 configured-nssai:
51   - sst: 0x01
52     sd: 0x010203
53
54 # Default Configured NSSAI for this UE
55 default-nssai:
56   - sst: 1
57     sd: 1
58
59 # Supported integrity algorithms by this UE
60 integrity:
61   IA1: true
62   IA2: false
63   IA3: false
64
65 # Supported encryption algorithms by this UE
66 ciphering:
67   EA1: false
68   EA2: false
69   EA3: false
70
71 # Integrity protection maximum data rate for user plane
72 integrityMaxRate:
```

```
73 uplink: 'full'  
74 downlink: 'full'
```