



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

José Luis Maziero Baretta

Encapsulamento de protocolos Ethernet Industrial utilizando a rede 5G

Blumenau
2022

José Luis Maziero Baretta

Encapsulamento de protocolos Ethernet Industrial utilizando a rede 5G

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. Dr. Adão Boava

Coorientador: Me. Christian Mailer

Blumenau

2022

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

Maziero Baretta, José Luis

Encapsulamento de protocolos Ethernet Industrial
utilizando a rede 5G / José Luis Maziero Baretta ;
orientador, Adão Boava, coorientador, Christian Mailer,
2022.

74 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Blumenau,
Graduação em Engenharia de Controle e Automação, Blumenau,
2022.

Inclui referências.

1. Engenharia de Controle e Automação. 2. 5G. 3.
Protocolos Ethernet Industrial. 4. Indústria 4.0. 5.
L2TPv3. I. Boava, Adão. II. Mailer, Christian. III.
Universidade Federal de Santa Catarina. Graduação em
Engenharia de Controle e Automação. IV. Título.

José Luis Maziero Baretta

Encapsulamento de protocolos Ethernet Industrial utilizando a rede 5G

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, 22 de dezembro de 2022.

Banca Examinadora:

Prof. Dr. Adão Boava
Universidade Federal de Santa Catarina

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

Prof. Dr. Mauri Ferrandin
Universidade Federal de Santa Catarina

Dedico este trabalho à minha família.
Com muita gratidão.

AGRADECIMENTOS

Agradeço à minha mãe Marivone, ao meu pai Josenei e à minha irmã Alícia. É e sempre vai ser por vocês.

Aos amigos Lucas, Rafael, Bruno, Carlos, Alexandre, Vitor, Adriano e tantos outros que frequentaram o apartamento 401 e dividiram comigo os melhores momentos da minha vida.

Estendo os agradecimentos ao meu amigo Luiz Soldatelli, pela parceira desenvolvida ao longo de todos os trabalhos da faculdade, e à minha amiga Laura, por todo o incentivo, principalmente durante a fase final da minha jornada acadêmica.

Por fim, agradecimentos especiais ao meu orientador, Prof. Dr. Adão Boava, e ao meu coorientador, Me. Christian Mailer, por todos os ensinamentos e aconselhamentos prestados durante o desenvolvimento deste trabalho.

RESUMO

A Indústria 4.0 é caracterizada por incorporar novas tecnologias ao ambiente de trabalho e proporcionar ganhos substanciais de produtividade, mas que trazem consigo novas demandas quanto às capacidades das rede de comunicação utilizadas. O 5G, por sua vez, é capaz de oferecer em uma de suas classes de uso, dentre muitas outras características, alta confiabilidade e baixíssima latência de comunicação. Neste trabalho, propôs-se uma maneira de integrar esses dois cenários ao aplicar uma técnica de encapsulamento de protocolos industriais baseados em *Ethernet*, bem como seu envio através da rede de acesso de rádio do 5G, se utilizando de ferramentas existentes e já consolidadas no contexto das redes de comunicação. Escolheu-se o L2TPv3 (*Layer 2 Tunnelling Protocol version 3*) — uma aplicação que implementa um protocolo de tunelamento de camada dois — para encapsular os protocolos industriais que implementam o *Ethernet* na camada de enlace. Para simular a rede de acesso de rádio do 5G, utilizou-se o pacote OpenAirInterface. Configurou-se *network namespaces* no sistema operacional Linux de modo a simular dispositivos comuns das redes industriais, como controladores, sensores e atuadores. Realizou-se, então, testes de integração, enviando pacotes encapsulados para serem transmitidos pelo 5G. Ao final, comprovou-se a eficácia do sistema desenvolvida através da análise dos dados coletados.

Palavras-chave: 5G; Protocolos Ethernet Industrial; Indústria 4.0; L2TPv3; OpenAirInterface.

ABSTRACT

Industry 4.0 is characterized by incorporating new technologies into the work environment and providing substantial productivity gains, but which bring with it new demands regarding the capabilities of the communication networks used. 5G, in turn, is capable of offering in one of its classes of use, among many other characteristics, high reliability and very low latency communication. In this work, a way to integrate these two scenarios was proposed by developing a technique that enables the encapsulation of industrial protocols based on Ethernet, as well as their transmission through the 5G radio access network, using existing tools that are already consolidated in the context of communication networks. L2TPv3 (Layer Two Tunneling Protocol Version 3) — an application that implements a layer two tunneling protocol — was chosen to encapsulate the industrial protocols that implement Ethernet in their physical layer. To simulate the 5G radio access network, the application OpenAirInterface was used. Network namespaces were configured in the Linux operating system in order to simulate common industrial network devices, such as controllers, sensors and actuators. Integration tests were performed, sending encapsulated packets to be transmitted over 5G. In the end, the effectiveness of the developed methodology was confirmed through the analysis of the collected data.

Keywords: 5G; Industrial Ethernet Protocols; Industry 4.0; L2TPv3; OpenAirInterface.

LISTA DE FIGURAS

Figura 1 – Arquitetura das redes de telecomunicações móveis.	19
Figura 2 – Exemplo de estação base.	20
Figura 3 – Exemplo ilustrativo do Modelo OSI.	21
Figura 4 – Requisitos da rede 5G e suas relações com os casos de uso.	29
Figura 5 – Arquitetura da rede de rádio 5G.	32
Figura 6 – Funções de rede essenciais do 5GC.	34
Figura 7 – Arquitetura de acesso de uma rede <i>non-3GPP</i> ao 5GC.	35
Figura 8 – Sessão PDU.	36
Figura 9 – <i>Software-Defined Networking</i>	38
Figura 10 – MU-MIMO e <i>beamforming</i>	40
Figura 11 – Classificação dos protocolos <i>Ethernet</i> Industrial.	42
Figura 12 – <i>Network namespace</i>	46
Figura 13 – <i>Network namespace</i> com as interfaces <i>veth</i>	47
Figura 14 – Encapsulamento e envio de pacotes no 5G utilizando túneis L2.	50
Figura 15 – Arquitetura A.	50
Figura 16 – Arquitetura B.	52
Figura 17 – Arquitetura B simplificada.	53
Figura 18 – Arquitetura A detalhada.	60
Figura 19 – Arquitetura B detalhada.	60
Figura 20 – Gráfico do tempo de encapsulamento em função do tamanho do pacote.	62

LISTA DE TABELAS

Tabela 1 – Principais características das redes LTE.	27
Tabela 2 – Requisitos técnicos do 5G.	30
Tabela 3 – Principais diferenças entre o rádio 4G e 5G.	31

LISTA DE ABREVIATURAS E SIGLAS

3GPP	3rd Generation Partnership Project
5GC	<i>5G Core</i>
5GS	<i>5G System</i>
6LoWPAN	<i>IPv6 over Low-Power Wireless Personal Area Networks</i>
AMF	<i>Access and Mobility Management Function</i>
AMPS	<i>Advanced Mobile Phone Service</i>
API	<i>Application Programming Interface</i>
AUSF	<i>Authentication Server Function</i>
BSS	<i>Base Station Subsystem</i>
CA	<i>Carrier Aggregation</i>
CDMA	<i>Code Division Multiple Access</i>
Cgroup	<i>Control Group</i>
CN	<i>Core Network</i>
D2D	<i>Device-to-Device Communications</i>
DC	<i>Dual Connectivity</i>
DECT	<i>Digital Enhanced Cordless Telecommunications</i>
DN	<i>Data Network</i>
DNS	<i>Domain Name Servers</i>
DOCSIS	<i>Data Over Cable Service Interface Specification</i>
EDGE	<i>Enhanced Data Rates for GSM Evolution</i>
eMBB	<i>enhanced Mobile Broadband</i>
eMTC	<i>enhanced Machine Type Communication</i>
eNB	<i>evolved NodeB</i>
EPC	<i>Evolved Packet Core</i>
EPS	<i>Evolved Packet System</i>
ETSI	European Telecommunication Standard Institute
E-UTRA	<i>Evolved Universal Terrestrial Radio Access</i>
E-UTRAN	<i>Evolved UMTS Terrestrial Radio Access Network</i>
FDMA	<i>Frequency Division Multiple Access</i>
GGSN	<i>Gateway GPRS Support Node</i>
gNB	<i>next-generation Node B</i>
GPRS	<i>General Packet Radio Services</i>
GSM	<i>Global Systems for Mobile Communications</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	<i>IP Multimedia Subsystem</i>

IMT	International Mobile Telecommunications
IN	<i>Intelligent Network</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
IPC	<i>Inter-Process Communication</i>
IPsec	<i>Internet Protocol Security</i>
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union Radiocommunication Sector
L2F	<i>Layer 2 Forwarding Protocol</i>
L2TP	<i>Layer 2 Tunneling Protocol</i>
L2TPv3	<i>Layer 2 Tunneling Protocol version 3</i>
LTE	<i>Long Term Evolution</i>
LTE-A	<i>LTE Advanced</i>
M2M	<i>Machine-to-Machine</i>
MAC	<i>Medium 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>
MTU	<i>Maximum Transmission Unit</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>
NF	<i>Network Function</i>
NFV	<i>Network Function Virtualization</i>
ng-eNB	<i>next-generation evolved Node B</i>
NG-RAN	<i>Next-Generation Radio Access Network</i>
NMS	<i>Network Management System</i>
NMT	<i>Nordic Mobile Telephone</i>
NR	<i>New Radio</i>
NSA	<i>Non-Stand Alone</i>
NSS	<i>Network Switching Subsystem</i>
OAI	OpenAirInterface
OFDMA	<i>Orthogonal Frequency Division Multiple Access</i>
OSI	<i>Open System Interconnection</i>

OT	<i>Operational Technology</i>
PCF	<i>Policy Control Function</i>
PDC	<i>Personal Digital Cellular</i>
PDU	<i>Protocol Data Unit</i>
PHS	<i>Personal Handy Phone System</i>
PID	<i>Process IDentification</i>
PLMN	<i>Public Land Mobile Network</i>
PPTP	<i>Point-to-Point Tunnelling Protocol</i>
PSA	<i>PDU Session Anchor</i>
PSTN	<i>Public Switched Telephone Network</i>
QoS	<i>Quality of Service</i>
RAN	<i>Radio Access Network</i>
REST	<i>Representational State Transfer</i>
RFC	<i>Request for Comments</i>
RTT	<i>Round-Trip Time</i>
SA	<i>Stand Alone</i>
SBA	<i>Service-Based Architecture</i>
SDF	<i>Service Data Flow</i>
SDN	<i>Software-Defined Networking</i>
SDU	<i>Service Data Unit</i>
SGSN	<i>Serving GPRS Support Node</i>
SIM	<i>Subscriber Identity Module</i>
SINR	<i>Signal-to-Interference-Plus-Noise Ratio</i>
SMF	<i>Session Management Function</i>
SMS	<i>Short Message Service</i>
SMSC	<i>Short Message Service Centre</i>
SU-MIMO	<i>Single User Multiple Input Multiple Output</i>
TACS	<i>Total Access Communication System</i>
TCP	<i>Transmission Control Protocol</i>
TDMA	<i>Time Division Multiple Access</i>
TE	<i>Terminal Equipment</i>
TFT	<i>Traffic Flow Template</i>
UDM	<i>Unified Data Management</i>
UDP	<i>User Datagram Protocol</i>
UE	<i>User Equipment</i>
UICC	<i>Universal Integrated Circuit Card</i>
UMTS	<i>Universal Mobile Telecommunication System</i>
UPF	<i>User Plane Function</i>
URLLC	<i>Ultra Reliable Low Latency Communications</i>

USIM	<i>Universal Subscriber Identity Module</i>
UTC	<i>Universal Time Coordinated</i>
UTS	<i>UNIX Time-Sharing System</i>
V2X	<i>Vehicle-to-Everything</i>
VAS	<i>Value Added Services</i>
VMS	<i>Voice Mail System</i>
VPN	<i>Virtual Private Network</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	PROBLEMATIZAÇÃO	16
1.2	OBJETIVOS GERAIS	17
1.3	OBJETIVOS ESPECÍFICOS	17
1.4	ESTRUTURA DO DOCUMENTO	18
2	REVISÃO DE LITERATURA	19
2.1	ARQUITETURA DAS REDES MÓVEIS	19
2.1.1	Modelo OSI e Protocolos de Comunicação	21
2.2	EVOLUÇÃO DAS REDES MÓVEIS	22
2.2.1	Primeira Geração - 1G	23
2.2.2	Segunda Geração - 2G	24
2.2.3	Terceira Geração - 3G	25
2.2.4	Quarta Geração - 4G	25
2.3	A QUINTA GERAÇÃO DA REDE MÓVEL - 5G	28
2.3.1	Rádio do 5G	29
<i>2.3.1.1</i>	<i>Arquitetura de Rádio do 5G</i>	<i>30</i>
2.3.2	Core do 5G	32
<i>2.3.2.1</i>	<i>Non-3GPP Interworking Function</i>	<i>34</i>
<i>2.3.2.2</i>	<i>Protocol Data Unit Session</i>	<i>35</i>
2.3.3	Tecnologias do 5G	37
<i>2.3.3.1</i>	<i>Network Function Virtualization</i>	<i>37</i>
<i>2.3.3.2</i>	<i>Software-Defined Networking</i>	<i>37</i>
<i>2.3.3.3</i>	<i>Network Slicing</i>	<i>38</i>
<i>2.3.3.4</i>	<i>Beamforming</i>	<i>39</i>
<i>2.3.3.5</i>	<i>MIMO</i>	<i>39</i>
2.4	DEFINIÇÕES DA SEXTA GERAÇÃO - 6G	40
2.5	PROTOCOLOS ETHERNET INDUSTRIAL	41
2.6	OPENAIRINTERFACE	43
2.7	L2TP	43
2.8	LINUX NAMESPACES	44
2.8.1	Process ID (PID) namespace	45
2.8.2	Network (net) namespace	45
2.8.3	Outros namespaces	47
3	SISTEMA PROPOSTO	49
3.1	ARQUITETURA A	50
3.2	ARQUITETURA B	51
3.3	METODOLOGIA DOS TESTES	54

4	RESULTADOS	58
5	CONCLUSÕES	63
	REFERÊNCIAS	64
	APÊNDICE A – Criação da Arquitetura A	68
	APÊNDICE B – Criação da Arquitetura B	70
	APÊNDICE C – <i>helpers functions</i>	72

1 INTRODUÇÃO

A Indústria 4.0, também chamada de Quarta Revolução Industrial, é caracterizada não só por ser um ambiente industrial onde se tem um alto nível de automação, mas também onde equipamentos e produtos precisam comunicar-se entre si e desenvolver atividades de forma autônoma, sem intervenção de operadores. Esses produtos, como robôs e veículos autônomos, ambientes de simulação com realidade aumentada e *Big Data Analytics*, requerem parâmetros de comunicação que supram às suas necessidades específicas, e que são conquistados por meio de características como uma alta taxa de transferência de dados e baixa latência de comunicação. No entanto, as tecnologias de redes sem fio presentes na indústria não acompanharam os requisitos que vieram junto das novas demandas. Visando solucionar esse problema, surge a quinta geração das redes móveis.

O 5G chega ao mercado não só como uma evolução das redes de telecomunicação móvel, mas como uma tecnologia que traz recursos exclusivos, fornecendo uma enorme gama de serviços que antes eram inimagináveis para redes de telecomunicação móvel e possibilitando o surgimento de soluções disruptivas. Trazendo um *core* evoluído e totalmente novo se comparado às gerações anteriores, estações base muito mais poderosas (que fazem o uso de ondas milimétricas e tecnologias como *beamforming*), e novos conceitos como as redes definidas por *software*, funções de rede virtualizadas, e fatiamento de rede, o 5G é capaz de oferecer uma solução adequada e que supre as demandas dessas novas tecnologias (PENTTINEN, 2019).

A quinta geração das redes móveis é flexível, segura, confiável, suporta altas taxas de velocidade na troca de informações, baixíssimas taxas de latência na comunicação e uma grande quantidade de dispositivos conectados ao mesmo tempo (ROMMER, 2020).

1.1 PROBLEMATIZAÇÃO

Para muitas das tecnologias presentes na indústria atual, e muitos outros casos de uso que surgirão com a Indústria 4.0, o tempo de resposta é um parâmetro de extrema importância. Apesar do surgimento constante de novos protocolos e do melhoramento dos padrões existentes, a latência de comunicação tem sido uma barreira significativa para o avanço de tais tecnologias.

As especificações iniciais do 5G, lançadas no *Release 15* do 3GPP (3rd Generation Partnership Project), definiram a arquitetura fundamental da quinta geração da rede móvel e abordaram casos de uso que visam atender as novas demandas do mercado. São eles: banda larga móvel aprimorada (eMBB), comunicações massivas do tipo máquina (mMTC) e comunicações de baixa latência ultra-confiáveis (URLLC) (3GPP, Acesso em: 20 nov. 2022). Esse último, garantindo latência de comunicação menor que 1 milissegundo e confiabilidade de 99,999% (COX, 2020).

Através dos lançamentos seguintes, definidos no *Release 16* e no *Release 17*, o 3GPP focou em desenvolver as especificações e tecnologias que serão utilizadas na implementação dos dois últimos casos citados. Entre elas, têm-se as sessões PDU do tipo *Ethernet*, um tipo de comunicação utilizado na rede 5G e feito especialmente para tratar do envio de pacotes *Ethernet* de forma nativa.

Atualmente no ambiente industrial, muitos dos protocolos utilizados se baseiam no *Ethernet* (ao implementá-lo na camada de enlace). Dentre os principais exemplos, pode-se destacar o Modbus, o EtherCAT e o PROFINET. Por conta disso, sessões PDU do tipo *Ethernet* se tornam ideais para o envio de informações por meio destes protocolos. No entanto, embora definida pelo 3GPP, esse tipo de sessão ainda não está disponível para uso geral; o que torna impossível a comunicação, de forma nativa, da rede 5G com o *Ethernet* Industrial (mais genericamente, com qualquer protocolo de camada 2 baseado no *Ethernet*).

Por conta disso, faz-se necessário o uso de técnicas de encapsulamento. Ao longo deste trabalho será desenvolvida uma solução que faz uso de uma técnica de encapsulamento já conhecida e utilizada no contexto de redes de comunicação, o L2TPv3.

1.2 OBJETIVOS GERAIS

Este trabalho tem como objetivo geral desenvolver uma solução que aplica uma técnica de encapsulamento para pacotes de protocolos industriais baseados em *Ethernet*, além de simular o envio desses pacotes por meio da rede de acesso de rádio do 5G.

1.3 OBJETIVOS ESPECÍFICOS

Como objetivos específicos deste trabalho, têm-se:

- Estudo da arquitetura das redes de telecomunicação móvel;
- Estudo aprofundado da rede móvel de quinta geração e de seus principais componentes e serviços oferecidos;
- Estudo e implementação de técnica de encapsulamento de protocolos industriais baseados em *Ethernet*;
- Configuração e simulação da rede de acesso de rádio do 5G através da aplicação OpenAirInterface;
- Envio de pacotes encapsulados por meio do 5G;
- Análise de performance e resultados da técnica de encapsulamento empregada.

1.4 ESTRUTURA DO DOCUMENTO

No Capítulo 2 têm-se uma breve introdução a conceitos essenciais para este trabalho, como a arquitetura das redes móveis, o modelo OSI (*Open System Interconnection*) e protocolos de comunicação. Além disso, um breve histórico das gerações da rede móvel até a chegada do 5G. De maneira mais detalhada, é apresentado o 5G, bem como suas principais tecnologias e componentes constituintes.

Os procedimentos utilizados a fim de obter-se os objetivos propostos, bem como a técnica desenvolvida, são detalhados no Capítulo 3.

No Capítulo 4 são apresentados os dados obtidos e é discutida, de forma breve, a performance da técnica empregada pelo autor.

Por fim, o Capítulo 5 realiza uma avaliação geral sobre os resultados obtidos, além de uma breve discussão a cerca da técnica utilizada para sanar o problema discutido.

2 REVISÃO DE LITERATURA

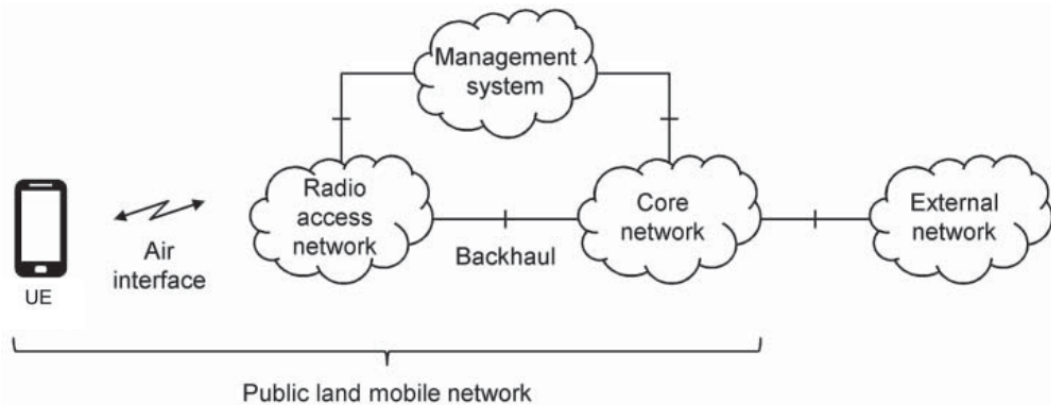
Nesse capítulo são introduzidos os conceitos principais das tecnologias utilizadas ao longo do trabalho. Primeiramente são apresentados conceitos básicos da arquitetura das redes de telecomunicação móvel, seguido da história da evolução de suas gerações, bem como as principais tecnologias utilizadas em cada uma delas. É apresentada de forma mais detalhada a quinta geração da rede móvel, o 5G, além de suas tecnologias chave. Estão presentes também outras tecnologias de grande importância para o desenvolvimento da solução proposta, que são: L2TPv3 (*Layer 2 Tunneling Protocol version 3*), OpenAirInterface e Linux *namespaces*.

2.1 ARQUITETURA DAS REDES MÓVEIS

Para um bom entendimento de tudo que será citado ao longo deste trabalho, é importante apresentar conceitos básicos da arquitetura da rede de telecomunicações.

Uma rede móvel de telecomunicações, usualmente gerenciada por uma operadora, é conhecida como PLMN (*Public Land Mobile Network*). Como pode ser visto na Figura 1, a arquitetura é composta, a grosso modo, por quatro elementos principais: o CN (*Core Network*), a rede de acesso de rádio, RAN (*Radio Access Network*), o NMS (*Network Management System*), e o equipamento do usuário, conhecido como UE (*User Equipment*) (COX, 2020).

Figura 1 – Arquitetura das redes de telecomunicações móveis.



Fonte: Adaptado de (COX, 2020).

O *core* transporta o tráfego entre o UE e a rede externa, que pode ser a Internet ou a PSTN (*Public Switched Telephone Network*). O *core* também é responsável por controlar a comunicação do UE com essas redes externas, além de armazenar informações sobre todos os assinantes da rede (COX, 2020).

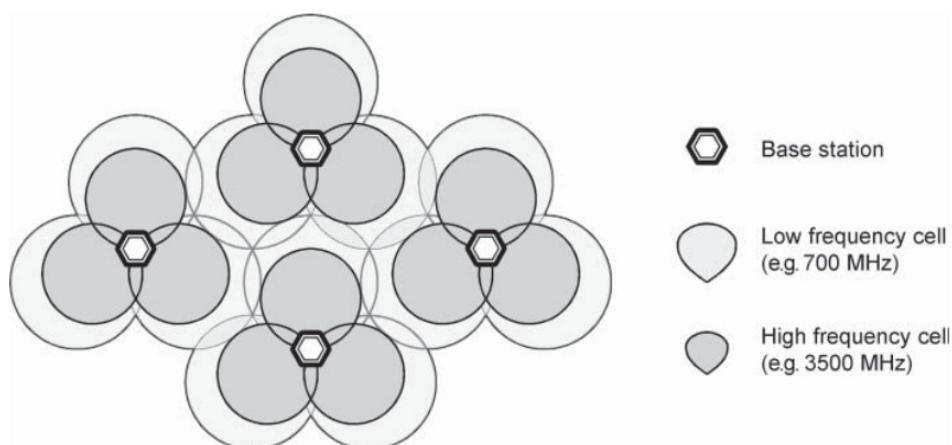
A RAN é responsável pela comunicação da rede de acesso com o UE. Ela se comunica com o *core* por meio de uma interface chamada *backhaul*, e com o UE pela interface de rádio aérea (COX, 2020).

Toda a rede é controlada pelo NMS, que oferece aos operadores e mantenedores a possibilidade de configurar o *core* e a RAN, bem como monitorar o seu funcionamento.

O UE é constituído por dois componentes. Um deles, o dispositivo responsável pela comunicação, é conhecido como ME (*Mobile Equipment*) (que em alguns casos pode também ser dividido em dois dispositivos distintos, o MT (*Mobile Termination*), que lida com todas as funções de comunicação, e o TE (*Terminal Equipment*), que é propriamente o equipamento) (TSG-T, Acesso em: 15 nov. 2022). O outro é o UICC (*Universal Integrated Circuit Card*), também conhecido como SIM (*Subscriber Identity Module*) Card, que é responsável pelo USIM (*Universal Subscriber Identity Module*), que tem como função armazenar dados específicos do usuário.

A RAN é formada por estações base. Uma estação base, exibida na Figura 2, pode transmitir e receber informações em uma ou mais frequências de rádio (que são definidas pela operadora). Em torno de cada frequência, o sinal de rádio ocupa uma faixa do espectro, chamada de largura de banda (como exemplo, uma estação base 5G transmitindo a uma frequência de 3,5 GHz, com uma largura de banda de 40 MHz, transmite na faixa de 3480 a 3520 MHz). Estações base de operadoras de telefonia móvel — quando se adquire uma licença por parte da operadora — têm permissão para transmitir em uma faixa definida; esta, presente no espectro licenciado (COX, 2020).

Figura 2 – Exemplo de estação base.



Fonte: (COX, 2020).

Cada estação base tem o poder de controlar uma ou mais células. Uma célula é definida como sendo uma pequena região geográfica composta por um grupo de canais de rádio que transmitem em uma frequência definida e com uma largura de banda específica

(DUSSUAX, Acesso em: 26 out. 2022).

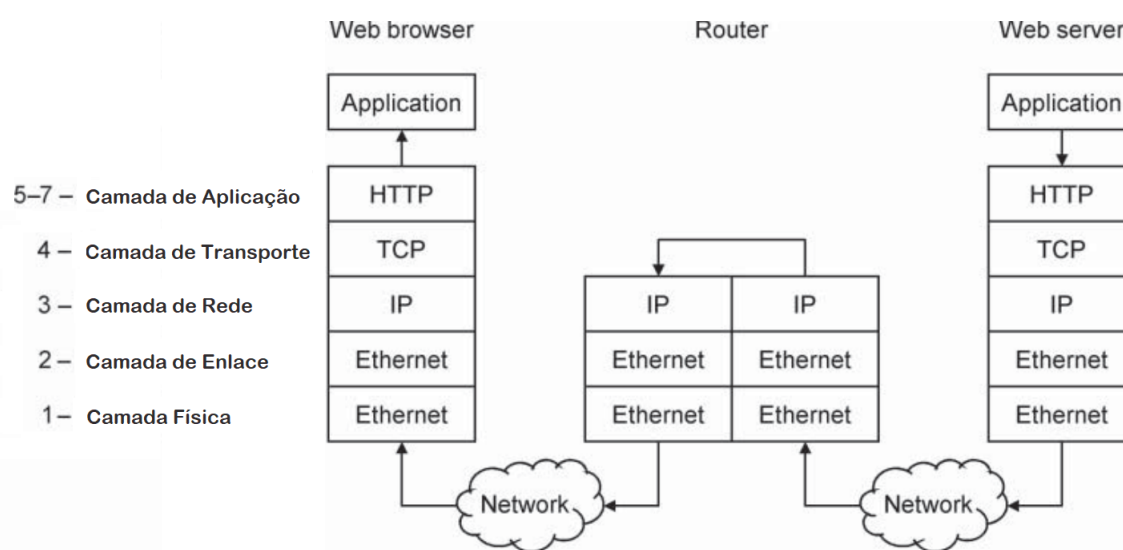
2.1.1 Modelo OSI e Protocolos de Comunicação

Outro conceito importante para se ter em mente, e fundamental para o entendimento de tecnologias utilizadas ao longo desse trabalho, como o L2TP (*Layer 2 Tunneling Protocol*), *Ethernet* e HTTP (*Hypertext Transfer Protocol*), é o modelo OSI (*Open System Interconnection*).

O OSI diz respeito a um modelo conceitual criado pela ISO (International Organization for Standardization) — com objetivo de ser um padrão a ser seguido por protocolos de comunicação — baseado no conceito de dividir um sistema de comunicação em camadas abstratas, empilhadas umas sobre as outras. Cada uma das camadas lida com um aspecto do processo de transmissão e recepção de informações.

Na obra (COX, 2020) é apresentada a Figura 3, que ilustra de forma didática as camadas do modelo OSI, e serve como um exemplo prático — e que será descrito a seguir — no qual um *webserver* manda informações ao navegador web de um usuário.

Figura 3 – Exemplo ilustrativo do Modelo OSI.



Fonte: Adaptado de (COX, 2020).

A camada de aplicação tem como função o registro de requisições. Aqui, um protocolo como o HTTP — protocolo base da *World Wide Web* — carrega informações do *webserver* e as passa para as camadas abaixo (de uma maneira que as informações sejam interpretáveis).

A camada de transporte é responsável pela comunicação fim-a-fim, em outras palavras, aplicações dessa camada só se comunicam com seu receptor. São dois os principais protocolos dessa camada. Um deles é o TCP (*Transmission Control Protocol*), que é um

protocolo orientado à conexão — o que significa que a transmissão dos dados só acontece se antes for estabelecida uma conexão (que é fechada após a transmissão dos dados), e que garante que os dados são entregues e livres de erro — e é adequado ao envio de informações que devem ser recebidas de forma confiável, como uma página web e um e-mail. O outro é o UDP (*User Datagram Protocol*), que envia os pacotes sem nenhum tipo de retransmissão (não garantindo assim que todas as informações serão recebidas e que não haverá erros), e que é adequado para comunicações sensíveis ao tempo, em que descartar pacotes é melhor do que esperar, como o *live streaming*.

Na camada de rede, o protocolo IP (*Internet Protocol*) coloca os pacotes na rota de destino correta. O responsável por esse processo são os roteadores — dispositivos que implementam apenas as três camadas mais baixas da pilha de protocolos —, que fazem isso ao inspecionar o endereço IP de destino.

A camada de enlace, nesse caso definida pelo protocolo *Ethernet*, gerencia a transmissão dos pacotes de um dispositivo ao outro, garantindo que ele seja re-transmitindo adiante caso ainda não tenha chegado ao seu destino final.

Por fim, a camada física lida com detalhes mais finos da comunicação, como a tensão do sinal transmitido.

Em todos os níveis da transmissão um protocolo recebe um pacote de dados proveniente do protocolo que se encontra na camada acima na forma de uma unidade de serviço de dados, ou SDU (*Service Data Unit*). Ele processa o pacote, adiciona um cabeçalho que descreve o processamento realizado e gera o resultado na forma de uma unidade de dados de protocolo, ou PDU (*Protocol Data Unit*). Esse processo se repete em todos os níveis do modelo OSI, onde a recém gerada PDU servirá como o SDU da próxima camada. Isso continua até que o pacote atinja a camada mais baixa de todas, momento em que ele é transmitido. O receptor inverte todas essas operações, usando os cabeçalhos que foram definidos durante a transmissão, de modo a desfazer o processamento feito pelo transmissor.

2.2 EVOLUÇÃO DAS REDES MÓVEIS

As redes de telecomunicações móveis são diferenciadas umas das outras através do termo gerações. Isso se dá, principalmente, pela grande diferença de complexidade encontrada nas tecnologias utilizadas para a implantação de cada uma delas. Os primeiros sistemas de telecomunicações móveis — analógicos e surgidos no início dos anos 1980, que ofereciam serviços de fala e tecnologias semelhantes — tinham como limitação, além da baixa quantidade de serviços oferecidos, falta de compatibilidade entre si.

A grande necessidade de serviços que atendessem a real demanda da comunicação móvel, e oferecessem mais compatibilidade entre tecnologias, resultou no surgimento de novas gerações da rede móvel. As organizações internacionais tiveram, nesse cenário, papel fundamental na concepção de um sistema que fornecesse melhoria nos serviços e fosse

mais transparente e compatível com as redes globalmente.

Entre as organizações que surgiram com o objetivo de regular e contribuir para a discussão e concepção das novas gerações, podem ser citadas, principalmente:

- **3GPP (3rd Generation Partnership Project):** fundado em 1998 por cinco empresas do ramo de tecnologia de comunicação móvel, é um órgão que foi originalmente criado com intuito de manter o controle do projeto e processo de especificação para as redes de terceira geração (MISHRA, 2004);
- **ITU (International Telecommunication Union):** com sede em Genebra, Suíça, é uma organização internacional que está presente dentro das Nações Unidas, onde os governos e o setor privado coordenam redes e serviços globais de telecomunicações. O ITU-R (International Telecommunication Union Radiocommunication Sector) é um dos três setores da organização e é responsável pelas comunicações de rádio (MISHRA, 2004);
- **ETSI (European Telecommunication Standard Institute):** é o órgão responsável por desenvolver os padrões da comunicação móvel na Europa. Foi criado inicialmente para desenvolver as especificações do GSM (*Global Systems for Mobile Communications*), mas também teve papel fundamental no desenvolvimento das redes de terceira geração, visto o sucesso comercial e técnico alcançado na geração anterior (MISHRA, 2007).

2.2.1 Primeira Geração - 1G

A primeira geração da rede móvel surgiu em meados da década de 1980 e usava técnicas de comunicação analógica, semelhantes aos rádios analógicos tradicionais (COX, 2020). A área geográfica era dividida em pequenas regiões, chamadas de células; daí a nomenclatura telefone celular (MISHRA, 2007).

O Japão tomou frente no desenvolvimento da tecnologia de comunicação móvel, fazendo com que as primeiras redes fossem implantadas em Tóquio. Nos anos seguintes surgiu o NMT (*Nordic Mobile Telephone*), que operava na Europa e foi o primeiro sistema de telefone celular totalmente automático. Outras tecnologias foram surgindo ao longo do tempo, como o AMPS (*Advanced Mobile Phone Service*), nos Estados Unidos, e o TACS (*Total Access Communication System*), no Reino Unido.

O AMPS foi o padrão de rede de acesso 1G mais amplamente implementado, sendo utilizado inclusive no Brasil. O MTSO (*Mobile Telephone Switching Office*) pode ser considerado o *core* dessa geração, pois realiza tarefas importantes, como *handover* e roteamento e gestão de ligações telefônicas (BOTH, C. ET AL, 2020), e interliga a PSTN (*Public Switched Telephone Network*) às estações base, que realizam as conexões usando modulação por frequência; além de realizar o controle de acesso ao meio por FDMA (*Frequency Division Multiple Access*) (DAVANÇO SILVA, G. H., Acesso em: 15 nov. 2022).

Na época, os padrões e regulamentações não eram bem estabelecidos, o que tornava as soluções existentes incompatíveis entre si — fazendo com que tecnologias como o *roaming*¹ fossem impossíveis de ser implementadas.

2.2.2 Segunda Geração - 2G

Devido as incompatibilidades e limitações das redes móveis da época, a ETSI iniciou uma série de discussões que resultaram no desenvolvimento de um mercado comum para os serviços e equipamentos de telecomunicações, visando a criação de um ambiente mais harmonizado. Com isso, surge o 2G, a segunda geração das redes móveis.

A segunda geração da rede móvel foi a primeira geração a contar com tecnologias digitais, o que trouxe um uso mais eficiente do espectro de rádio, possibilitando, assim, um maior número de usuários conectados, além da introdução de dispositivos menores e mais baratos (e que ajudaram a popularizar ainda mais as tecnologias de comunicação móvel). Entre os padrões dessa geração, estão: TDMA IS-136, CDMA IS-95, PDC (*Personal Digital Cellular*), PHS (*Personal Handy Phone System*), além do mais popular de todos, o GSM (COX, 2020).

- **GSM:** é dividido em duas camadas principais, a BSS (*Base Station Subsystem*) e a NSS (*Network Switching Subsystem*). A rede é capaz de oferecer todos os serviços básicos, como fala e troca de dados (com taxa de transmissão de até 9,6 kbps), fax, além da possibilidade de conexão com a rede de telefonia fixa (MISHRA, 2004).
- **VAS (*Value Added Services*):** a primeira evolução da rede GSM veio com a adição de duas novas plataformas: VMS (*Voice Mail System*) e SMSC (*Short Message Service Centre*). O SMS (*Short Message Service*) se provou tão popular que em algumas redes constituía a maior parte do tráfego. Junto do VAS, o IN (*Intelligent Network*) também marcou a tecnologia GSM, trazendo às operadoras a chance de criar uma nova gama de serviços, que incluía sistemas antifraude e o sistema pré-pago (MISHRA, 2004).
- **GPRS (*General Packet Radio Services*):** visando atender novos assinantes móveis e a demanda de altas taxas de *bits* para transmissão de dados, surge o GPRS. Os principais elementos dessa infraestrutura são o SGSN (*Serving GPRS Support Node*) e o GGSN (*Gateway GPRS Support Node*) (TUDE, Acesso em: 05 nov. 2022). Essas novas tecnologias trouxeram suporte ao roteamento IP, servidores de *firewall* e DNS (*Domain Name Servers*); o que proporcionou acesso *wireless* à Internet, com taxa de transmissão de dados chegando a 150 kbps (MISHRA, 2004).

¹ *Roaming* de dados é uma tecnologia que permite ao usuário enviar e receber dados através de redes móveis, mesmo fora da área de cobertura da sua operadora.

- **EDGE (*Enhanced Data Rates for GSM Evolution*)**: com suporte à voz e dados já atendidos, a necessidade foi a de aumentar as taxas de transmissão. Isso foi possível com a introdução do EDGE, que é considerado um subconjunto do GPRS, pois pode ser instalado em qualquer sistema que tenha o GPRS implantado (MISHRA, 2007). Usando novas tecnologias de modulação, o GPRS elevou as taxas de transmissão de dados a 384 kbps (MISHRA, 2004).

Toda a família do sistema 2G ainda está operacional hoje em dia. No entanto, a segunda geração da rede móvel carecia em relação à capacidade de usuários conectados, suporte a *roaming* global, e na qualidade e quantidade de tráfego de dados oferecida. Tudo isso direcionou a indústria a desenvolver um sistema que tivesse maior suporte global, marcando assim, o início da terceira geração.

2.2.3 Terceira Geração - 3G

Por conta da melhoria dos serviços oferecidos, da ampliação da implantação global e do aumento de interesse do público geral, novas demandas surgiram. Isso culminou na criação da terceira geração da rede móvel, que teve como objetivo principal oferecer altas taxas de transmissão para serviços multimídia.

Os requisitos comerciais do 3G foram definidos na iniciativa IMT-2000, de responsabilidade da ITU. Nela, destaca-se como esperado à uma rede 3G características como (VIEIRA PEIXOTO, Acesso em: 26 out. 2022):

- Taxas de transmissão:
 - 144 kbps via satélites e áreas ao ar livre no ambiente rural;
 - 384 kbps em áreas ao ar livre no ambiente urbano;
 - 2048 kbps em curtas e médias distâncias;
- Melhorias nos serviços de multimídia;
- Usabilidade em todas as modalidades populares (como telefone celular, e-mail, fax ou vídeo conferência);
- Flexibilidade de roteamento (através de repetidores, satélites, LAN);
- Capacidade de *roaming* por toda a Europa, Japão e América do Norte.

UMTS (*Universal Mobile Telecommunication System*), CDMA2000, DECT (*Digital Enhanced Cordless Telecommunications*), WiMAX (*Worldwide Interoperability for Microwave Access*), e algumas variações do GSM como o EDGE, foram alguns dos padrões capazes de cumprir com os requisitos estabelecidos na IMT-2000 (PENTTINEN, 2019).

2.2.4 Quarta Geração - 4G

A popularização dos dispositivos móveis de usuários — especialmente dos smartphones e tablets — ocorrida no final da primeira década deste século, impulsionou a indústria

de telecomunicações. Esses dispositivos passaram a contar com melhores recursos computacionais, o que contribuiu extensivamente para o crescimento da demanda de banda larga móvel, com altas taxas de dados e QoS (*Quality of Service*)². Essa grande demanda alavancou o surgimento da rede móvel de quarta geração, o 4G.

Apresentado em 2008 pelo ITU-R, o IMT-Advanced definiu as capacidades mínimas esperadas de um sistema 4G (BOTH, C. ET AL, 2020):

- Ser integralmente baseado em comutação por pacotes IP;
- Interoperabilidade com os padrões existentes;
- Suportar taxas de dados de pico de até aproximadamente 100 Mbps para acesso móvel de alta mobilidade, e 1 Gbps para acesso móvel de baixa mobilidade;
- Incrementar a capacidade de atendimento de usuários por célula com uso de recursos de rede compartilhados;
- Suportar a conexão simultânea a duas ou mais células durante uma conexão para oferecer transição suave em redes heterogêneas;
- Suportar aplicações multimídia de próxima geração de alta qualidade.

O ITU-R concluiu a avaliação de seis candidatos, submetidos globalmente, para compor o hall de tecnologias que se adequavam às normas do 4G (PENTTINEN, 2019). No entanto, apenas duas das tecnologias propostas foram aceitas: o LTE (*Long Term Evolution*), desenvolvido pelo 3GPP, e o IEEE 802.16³, desenvolvido por um grupo de trabalho do IEEE (Institute of Electrical and Electronics Engineers), que também ganhou o nome WiMAX (BOTH, C. ET AL, 2020).

Os dois padrões empregaram novas técnicas na sua interface aérea: OFDMA (*Orthogonal Frequency Division Multiple Access*) e MIMO (*Multiple Input Multiple Output*). Nesse geração, o WiMAX assumiu um importante papel como tecnologia de acesso sem fio de banda larga fixa, enquanto o LTE se tornou o padrão universal para a interface aérea do 4G.

No padrão LTE, a estação rádio base é conhecida como eNB (*evolved NodeB*), a rede de acesso é a E-UTRAN (*Evolved UMTS Terrestrial Radio Access Network*) e o *core* é formado pelo EPC (*Evolved Packet Core*). Oficialmente, todo o sistema é conhecido como EPS (*Evolved Packet System*) (COX, 2020).

A indústria de telecomunicação móvel considera a primeira versão do LTE como 4G, mas, na verdade, se olharmos as suas especificações técnicas iniciais, como mostrado na Tabela 1, fica claro que ele não cumpre os requisitos mínimos impostos pelo IMT-Advanced. A explicação prática para essa consideração se dá pelo fato de que o LTE está

² QoS (*Quality of Service*) é obtido através de um conjunto de tecnologias presentes em uma rede que funcionam com intuito de controlar o tráfego e garantir o desempenho de aplicativos essenciais com capacidade de rede limitada.

³ Os estudos a cerca do IEEE 802.16 foram motivados pelo enorme sucesso do IEEE 802.11 (popularmente conhecido como WiFi), desenvolvido para a comunicação sem fio em redes locais (BOTH, C. ET AL, 2020).

muito mais próximo do 4G que do 3G, tanto nas suas especificações quanto nas tecnologias utilizadas (além disso, o mercado adotou o nome pois não fazia sentido se amarrar a um nome antigo). Em contrapartida, a resposta do 3GPP veio através de novas especificações para uma versão aprimorada do LTE, conhecida como LTE-A (*LTE Advanced*).

O LTE-A (*LTE Advanced*) trouxe ganhos significativos de capacidade em relação ao LTE. O aprimoramento mais importante foi o CA (*Carrier Aggregation*). Essa técnica traz um aumento significativo da taxa de transmissão de dados pois permite que o usuário se comunique com múltiplas células de rádio ao mesmo tempo, que transmitem em faixas de frequências distintas, mas ainda são controladas por uma única eNB. Outro avanço foi a introdução de uma tecnologia chamada DC (*Dual Connectivity*), que permite que as células sejam controladas por duas estações base ao invés de uma (essa tecnologia foi muito importante posteriormente para a introdução do 5G) (COX, 2020).

Os aprimoramentos seguintes do LTE são conhecidos como LTE-Advanced Pro, ao qual também se deu o nome 4.5G (BOTH, C. ET AL, 2020). Essa nova tecnologia trouxe taxas de transmissão consideravelmente mais altas (cerca de 3 vezes mais altas que o LTE-A) (BOTH, C. ET AL, 2020), praticamente dobrou a capacidade da rede sem a necessidade de espectro adicional ou novas estações base (através da tecnologia MIMO), diminuiu significativamente a latência e aumentou a drasticamente a eficiência energética (COX, 2020).

Tudo isso permitiu a introdução de novos casos de uso pra tecnologia LTE, como: eMTC (*enhanced Machine Type Communication*), D2D (*Device-to-Device Communications*) e Narrowband IoT; que deixaram o 4G em um alinhamento muito mais próximo em relação aos melhoramentos futuros propostos pela rede 5G (PEREIRA MARTINS, A.; OLIVEIRA, F. C. DE, Acesso em: 15 nov. 2022).

Na Tabela 1 é apresentado um comparativo dos três padrões citados acima e definidos pela 3GPP da tecnologia LTE.

Tabela 1 – Principais características das redes LTE.

Tecnologia	LTE	LTE-Advanced	LTE-Advanced Pro
Downlink	100 Mbps	1,0 Gbps	3,0 Gbps
Uplink	50 Mbps	0,5 Gbps	1,5 Gbps
Canalização (MHz)	20	100	640
Latência (ms)	~10	~10	<2

Fonte: Adaptado de (TELECO, Acesso em: 04 nov. 2022).

2.3 A QUINTA GERAÇÃO DA REDE MÓVEL - 5G

A ITU começou, em 2012, a definir o sucessor do IMT-Advanced (apresentado na Subseção 2.2.4), o qual deu-se o nome IMT-2020, e que viria a servir como um ponto de partida para o início das pesquisas relacionados à rede móvel de quinta geração, o 5G. Essa movimentação por parte da ITU também estimulou estudos vindos do 3GPP, visando desenvolver uma tecnologia que se adequasse aos requisitos do IMT-2020.

O 5G, diferente de todas as outras gerações passadas, é uma evolução do padrão anterior em aspectos que vão além da melhora da latência e da taxa de dados. Ele chega não só apenas com o intuito de melhorar o 4G, mas de fornecer uma enorme gama de serviços que antes eram inimagináveis para uma rede de telecomunicações móvel. Mais do que apenas melhorar a largura de banda e reduzir a latência, o 5G permitirá que soluções verdadeiramente disruptivas surjam em todos os tipos de setores.

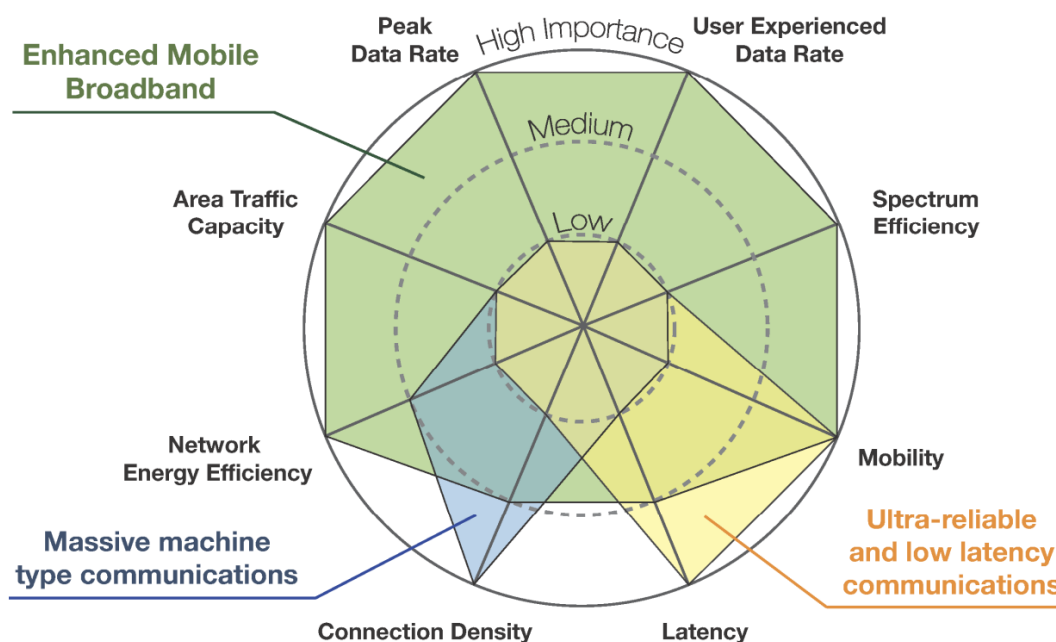
Os estudos relacionados à implantação do 5G levaram à divisão da tecnologia em três casos de uso principais, que estão expostos abaixo e também podem ser vistos na Figura 4.

- **eMBB (*enhanced Mobile Broadband*):** é a categoria onde o 5G vai entregar velocidade na casa dos *gigabits*. É voltado para aplicações que exigem altas taxas de dados em áreas densamente povoadas;
- **mMTC (*massive Machine Type Communications*):** é caracterizado pela agregação e análise de dados de um grande número de dispositivos conectados, possibilitando, assim, aplicações do tipo M2M (*Machine-to-Machine*) e IoT (*Internet of Things*). Nesse caso de uso as mensagens normalmente possuem tamanho reduzido, pois contém apenas dados de medições ou comandos para atuadores (MAILER, C, Acesso em: 10 nov. 2022). Os dispositivos devem ser baratos e exigem baixo consumo de energia para garantir longa duração da bateria;
- **URLLC (*Ultra Reliable Low Latency Communications*):** é voltada para serviços que exijam latências muito baixas e alta confiabilidade. Será essencial para a Indústria 4.0 e outras aplicações, como direção autônoma e assistida, missões críticas e *drones*.

No que segue as especulações do mercado, é esperado que o 5G se estabeleça verdadeiramente tendo como base os dois últimos casos apresentados. Embora o eMBB, à primeira vista, pareça ser a tecnologia mais atrativa oferecida pelo 5G, as novas vertentes e modelos de negócios que serão explorados a partir de aplicações mMTC e URLLC, como carros autônomos, Indústria 4.0 e realidade virtual, parecem ser peça-chave para a adoção global dessa tecnologia.

Há também outro caso de uso que – embora não reconhecido pelo ITU – é importante para o 5G. Trata-se do V2X (*Vehicle-to-Everything*), que foi definido pelo 3GPP e

Figura 4 – Requisitos da rede 5G e suas relações com os casos de uso.



Fonte: (ITU-R, Acesso em: 20 nov. 2022).

abrange as trocas de dados de veículos com a rede de telecomunicação móvel, com outros veículos e com pedestres que estão nas proximidades (COX, 2020).

Na Tabela 2 são mostradas as especificações do 5G, definidas pela 3GPP, que se adequam as definições estabelecidas pelo ITU-R no IMT-2020; além de compará-las com as especificações da última geração, o 4G LTE, e associá-las as novas classes de uso.

O 5GS (*5G System*) é composto por dois componentes principais, são eles: o acesso via rádio, chamado de NG-RAN (*Next-Generation Radio Access Network*), que suporta o NR (*New Radio*), e o 5GC (*5G Core*). Ambos evoluíram consideravelmente em comparação com as gerações passadas e serão detalhados nas Subseções 2.3.1.1 e 2.3.2, respectivamente.

2.3.1 Rádio do 5G

O 3GPP definiu no *Release 14*, de junho de 2016, as primeiras especificações do rádio do 5G, ao qual deu-se o nome NR (*New Radio*).

Muitos são os recursos exclusivos do NR em comparação ao LTE, como a ampla gama de frequências de operadora, várias opções de implantação (como possibilitar a escolha de cobertura por estação base) e os casos de uso (apresentados na Seção 2.3) que vão além do serviço de banda larga móvel (ZAIDI A.; BALDEMAIR, Acesso em: 20 nov. 2022). A Tabela 3 mostra as diferenças entre as principais características do 4G e do 5G.

Para atender as especificações do 5G, foram definidas duas faixas de frequência de rádio: a primeira se estende de 410 Mhz até 7.125 Mhz (chamada de FR1, *Frequency Range*

Tabela 2 – Requisitos técnicos do 5G.

Parâmetro	4G LTE-Advanced	5G	Principal caso de uso
Taxa de dados	500 Mbps (UL) 1000 Mbps (DL)	10 Gbps (UL) 20 Gbps (DL)	eMBB
Eficiência energética	n/a	Máxima	eMBB
Máx. velocidade do UE	350 km h^{-1}	500 km h^{-1}	eMBB
Latência	5 ms	0,5 ms	URLLC
Confiabilidade	n/a	99,999%	URLLC
Vida útil da bateria	10 anos	10-15 anos	mMTC
Densidade de conexão	60.000 km $^{-2}$	1.000.000 km $^{-2}$	mMTC

Fonte: Traduzido de (COX, 2020).

1 ou *sub-6* GHz), e a segunda de 24,25 GHz até 52,6 GHz (chamada de FR2, *Frequency Range 2* ou *mmWave*) (3GPP, E., Acesso em: 10 nov. 2022).

A escolha por altas frequências de rádio se justifica pela busca por uma largura de banda maior e alta taxa de dados. Em contrapartida, ondas *mmWave* trazem consigo problemas de cobertura. Visando mitigar isso, o 5G faz uso de tecnologias como o MIMO e o *beamforming*, que serão descritas ao longo da Subseção 2.3.3.

A decisão em torno das frequências que serão utilizadas ao longo do globo fica a cargo do país em que a implementação será feita e de suas agências reguladoras. Para ajudar nesse processo a ITU dividiu o globo em três regiões: a primeira região cobre a Europa, África e norte e oeste da Ásia (incluindo o Oriente Médio e a antiga União Soviética); a segunda região cobre as Américas; a terceira região abrange o restante da Ásia (incluindo Índia e China) e a Australásia⁴ (COX, 2020).

2.3.1.1 Arquitetura de Rádio do 5G

A implantação das redes de rádio 5G vem sendo feita através de duas arquiteturas distintas: a NSA (*Non-Stand Alone*), onde o 5G se beneficia da infraestrutura da rede 4G pré-existente – o que limita, obviamente, as tecnologias oferecidas e restringe o suporte apenas ao caso de uso eMBB – e a SA (*Stand Alone*), popularmente conhecida no Brasil

⁴ Australásia é a região que inclui a Austrália, a Nova Zelândia, a Nova Guiné e algumas ilhas menores da parte oriental da Indonésia.

Tabela 3 – Principais diferenças entre o rádio 4G e 5G.

Característica	4G LTE	5G NR
Codificação do canal de dados	Turbo	LDPC
Codificação do canal de controle	TBCC	Polar
Esquema de modulação para o <i>uplink</i>	SC-FDMA (Single-carrier frequency-division multiplexing)	DFT-S-OFDM; OFDM (opcional)
Esquema de modulação para o <i>downlink</i>	OFDM	OFDM
Largura de banda (MHz)	1,4; 3; 5; 10; 15; 20	5, ..., 100 (sub 6 GHz); 50, ..., 400 (above 6 GHz)
Espaçamento da subportadora (kHz)	15 (unicast, Multimedia Broadcast Multicast Service [MBMS]); 7,5/1,25 (portadora MBMS dedicada)	30, 60, 120; 240 (não para dados)
Máxima agregação de portadora (CC)	32	16
Máximo de portas de antena MIMO	8 (SU-MIMO); 2 (MU-MIMO);	8 (SU-MIMO); 16 (MU-MIMO);

Fonte: (MAILER, C, Acesso em: 10 nov. 2022).

como "5G Puro", onde a rede 5G funciona de maneira plena (sem fazer uso de nenhuma infraestrutura do 4G) e é capaz de oferecer suporte a todos os casos de uso apresentados na anteriormente Seção 2.3.

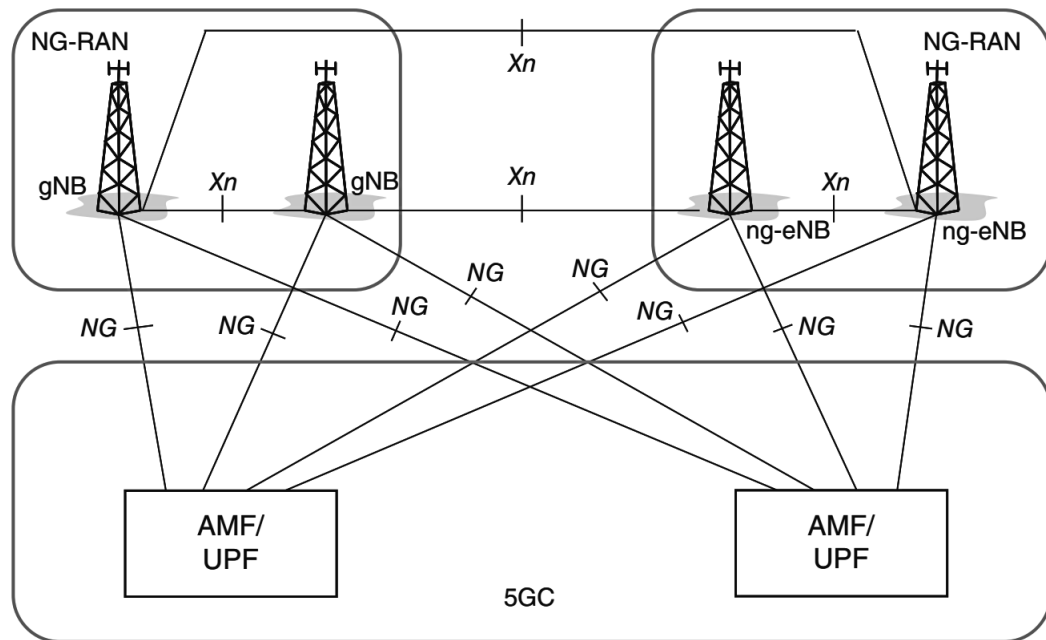
Ainda sobre os modelos de arquitetura adotados, pode-se citar alguns motivos que levam à escolha da opção NSA (que vão além da gestão do espectro, e entram no âmbito comercial), entre eles: a implantação rápida do 5G com investimento reduzido e a maximização de valor da infraestrutura já existente do 4G. A adoção do 5G SA, no entanto, traz para a operadora liderança tecnológica; o que pode, de certa forma, ajudar na prospecção de novos clientes que queiram utilizar a rede sem estarem limitados ao caso de uso eMBB (NOLL BARRETO, Acesso em: 10 nov. 2022).

Tendo em mente as arquiteturas citadas anteriormente: para a estação base nativa do 5G, dá-se o nome de gNB (*next-generation Node B*). Quando a estação base for LTE (proveniente da infraestrutura atual do 4G), o termo utilizado é ng-eNB (*next-generation evolved Node B*). De forma genérica, pode-se referir às estações base 5G como NG-RAN, independente de serem gNB ou ng-eNB.

A arquitetura de rede de rádio do 5G é composta por múltiplas estações base conectadas entre elas mesmas e ao 5GC, como visto na Figura 5. As estações base são

interconectadas pelas interfaces Xn (estas divididas em Xn-C, a interface de sinal, e Xn-U, a interface para transferência de dados). As interfaces de comunicação com os elementos do *core* AMF (*Access and Mobility Management Function*) e UPF (*User Plane Function*) – ambas funções de rede virtualizadas, detalhadas na Seção 2.3.2 – são, respectivamente, N2 e N3 (MAILER, C, Acesso em: 10 nov. 2022).

Figura 5 – Arquitetura da rede de rádio 5G.



Fonte: (PENTTINEN, 2019).

Dentre as funcionalidades do rádio 5G, as principais são: gerenciamento de recursos de rádio, gerenciamento de cabeçalho IP, gerenciamento da AMF, funcionalidade de roteamento, configuração e liberação de conexões, funcionalidade de agendamento, medições, marcações de pacotes, gerenciamento de sessões, fatiamento de rede, funções de QoS, suporte de UEs em estado RRC_INACTIVE, função de distribuição para mensagens NAS (*Non-Access Stratum*), compartilhamento de rede de acesso de rádio, DC e interfuncionamento entre NR e E-UTRA (*Evolved Universal Terrestrial Radio Access*), a interface de rádio do E-UTRAN (MAILER, C, Acesso em: 10 nov. 2022).

2.3.2 Core do 5G

O *core*, juntamente com o NG-RAN, é um dos elementos essenciais do 5G, como já dito na Seção 2.3. No 5G, o conceito principal trazido pelo *core* é uma arquitetura baseada em serviços, que, de maneira resumida, provê uma estrutura flexível em que suas funcionalidades são divididas em nós, chamados de NF (*Network Function*), ligados a um

barramento comum, e que podem interagir entre si de maneira independente (MAILER, C, Acesso em: 10 nov. 2022).

Uma arquitetura baseada em serviços permite a utilização de recursos em nuvem, orquestração e *Edge Computing* (MAILER, C, Acesso em: 10 nov. 2022), para escalonar a rede e adaptá-la conforme variações nas cargas de processamento, trazendo maior eficiência e estabilidade.

O *core* do 5G foi definido e concebido de modo que a implementação e substituição de outras tecnologias seja possível. Além disso, ele possui suporte a tecnologias *multivendor* de maneira mais facilitada se comparado com as gerações anteriores (PENTTINEN, 2019).

Por ter uma arquitetura baseada em serviços, as funções de rede que incluem lógica e funcionalidade para processar fluxos de sinalização não são interconectadas através de interfaces ponto-a-ponto, ao invés disso, elas disponibilizam serviços que serão utilizados pelas outras funções de rede (ROMMER, 2020). Para cada interação e troca de informações entre funções de rede, diz-se que uma delas age como produtora, e a outra como consumidora do serviço (ROMMER, 2020).

De forma mais detalhada, a arquitetura baseada em serviços faz com que cada função de rede ofereça um ou mais serviços para as outras funções presentes na rede. Na arquitetura do 5G esses serviços são disponibilizados por meio de interfaces de função de rede conectadas à SBA (*Service-Based Architecture*) e acessadas por meio de uma API (*Application Programming Interface*) que segue o modelo REST (*Representational State Transfer*)⁵ (ROMMER, 2020).

A Figura 6 mostra as principais funções de rede do 5GC (*5G Core*). Cada uma delas é implementada através de *software*, na forma de funções de rede virtualizadas, e são executadas em *hardware* genérico.

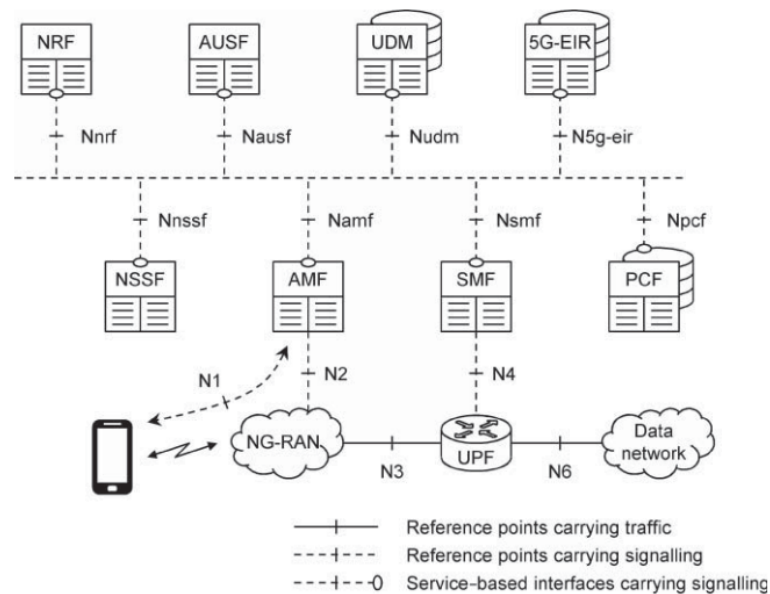
Através do 5GC um usuário pode se conectar a uma ou mais rede de dados, as chamadas DN (*Data Network*), que podem ser a Internet, IMS (*IP Multimedia Subsystem*), que fornece serviços como texto, voz, e vídeo) ou uma rede industrial privada de baixa latência. Cada conexão está associada a uma SMF (*Session Management Function*).

A SMF gerencia a conexão entre o UE e a DN, comunica-se com o UE por meio de mensagens de gerenciamento de sessão e ainda envia informações sobre o uso de dados do celular para o sistema de cobrança (COX, 2020). A SMF ainda controla uma ou mais UPF (*User Plane Function*), que é responsável por gerenciar o tráfego do usuário através de N4 (MAILER, C, Acesso em: 10 nov. 2022).

Cada UE é servido por uma AMF (*Access and Mobility Management Function*). A AMF controla as interações do UE com o 5GC por meio da sinalização de gerenciamento de mobilidade e retransmite a sinalização de gerenciamento de sessão entre o UE e seus respectivos SMFs. Ambos os conjuntos de mensagem são retransmitidos pela NG-RAN

⁵ REST (*Representational State Transfer*) é um design de arquitetura construído para servir aplicações *web*. Uma API que segue o modelo REST está em conformidade com as restrições dessa arquitetura.

Figura 6 – Funções de rede essenciais do 5GC.



Fonte: (COX, 2020).

através de N1, enquanto a AMF se comunica com o próprio NG-RAN por N2 (COX, 2020).

A UDM (*Unified Data Management*) atua como repositório central para informações relacionadas aos assinantes, apresentando as informações para as outras funções na rede. A PCF (*Policy Control Function*) fornece à SMF políticas relacionadas ao gerenciamento da sessão, como por exemplo a taxa máxima de dados, e à AMF políticas relacionadas ao gerenciamento de mobilidade (COX, 2020).

A NSSF (*Network Slice Selection Function*) determina as fatias de rede as quais o UE será registrado, além de atribuir uma AMF – que suporte essas fatias – para lhe servir. Por fim, a AUSF (*Authentication Server Function*) ajuda a AMF durante o procedimento de autenticação do UE (COX, 2020).

Além disso, o *core* suporta tanto o acesso por redes 3GPP, quanto *non-3GPP*. Isso permite, por exemplo, comunicação por meio de uma rede WiFi. A Subseção 2.3.2.1 explica como isso é feito.

2.3.2.1 *Non-3GPP Interworking Function*

A função de rede N3IWF (*Non-3GPP Interworking Function*), vista na Figura 7, permite a integração do *core* 5G com redes de acesso não definidas pela 3GPP, como o IEEE 802.11 WiFi e o DOCSIS (*Data Over Cable Service Interface Specification*).

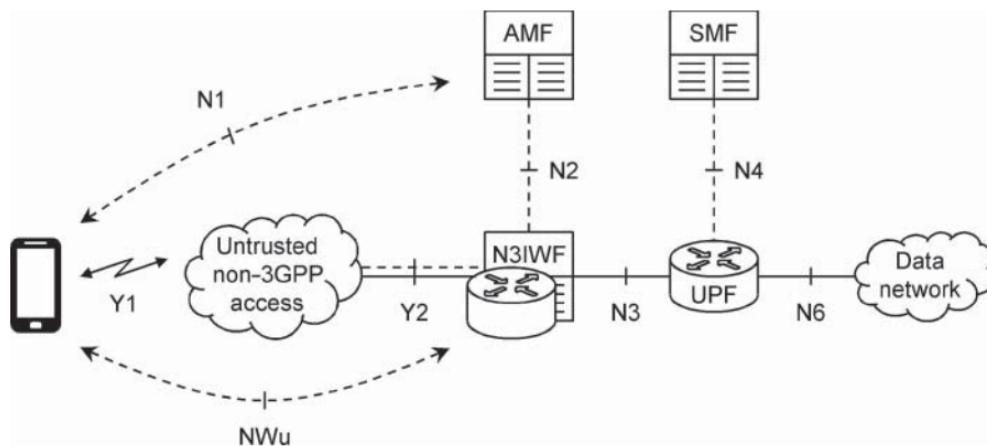
A N3IWF é responsável por reencaminhar sinalização e dados entre o *core* e a rede de acesso *non-3GPP* (BOTH, C. ET AL, 2020). Todo o tráfego a partir da N3IWF é

transmitido através de canais seguros e isolados dos demais tráfegos 3GPP. Esse isolamento ocorre tanto para o tráfego de dados, quanto para o tráfego de controle (inclusive com o que acontece antes do processo de autenticação) (BOTH, C. ET AL, 2020).

A conexão de um dispositivo com uma rede *non-3GPP* é servida por uma AMF (assim como no caso de uso padrão, quando através da NG-RAN). A autenticação é feita pela AMF e pela AUSF. A AMF se comunica com o dispositivo através de N1, e com o acesso através de N2 (MAILER, C, Acesso em: 10 nov. 2022).

A N3IWF estabelecida comunica-se com o *core* através de N2 e N3. Cria-se então um túnel IPsec entre o dispositivo e a N3IWF, através do qual sinais e dados entre dispositivo e rede serão encaminhados. Esse túnel recebe o nome NWu (ROMMER, 2020).

Figura 7 – Arquitetura de acesso de uma rede *non-3GPP* ao 5GC.



Fonte: (COX, 2020).

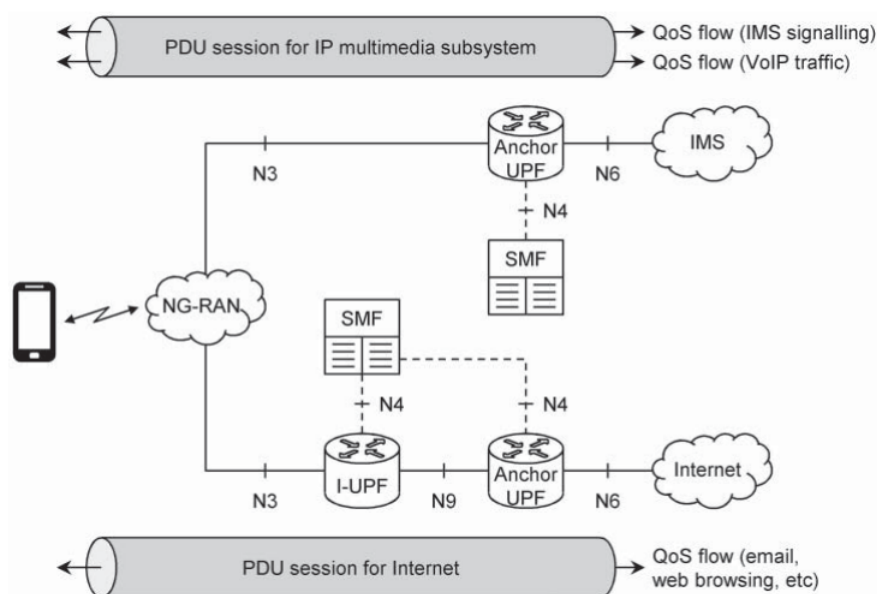
2.3.2.2 Protocol Data Unit Session

À conexão do UE com uma rede de dados externa se dá o nome de sessão PDU. Assim que estabelecida, cada sessão PDU é associada com pelo menos uma UPF. Denomina-se PSA (*PDU Session Anchor*) a ligação entre o 5GC e uma DN, que é feita através da interface N6. Pode-se haver também UPFs intermediárias, chamados de I-UPFs, que encaminham o tráfego de dados entre as âncoras e o NG-RAN (COX, 2020).

Como observado na Figura 8, o *backhaul* de cada sessão PDU está associado a uma única UPF, que pode desempenhar o papel de uma PSA, de uma I-UPF, ou de ambas (COX, 2020).

Uma sessão PDU pode lidar com três tipos de fluxos, chamados de *flow*: os de pacotes, os de dados de serviços – definidos como SDF (*Service Data Flow*) –, e os de QoS (*Quality of Service*) (MICROSOFT, Acesso em: 20 nov. 2022).

Figura 8 – Sessão PDU.



Fonte: (COX, 2020).

Uma sessão PDU contém um ou mais fluxos QoS (isso depende do tipo de conexão). Cada fluxo é um conjunto de dados que carrega parâmetros de conexão como: taxa de dados, taxa de erro e atraso de ponta a ponta (COX, 2020).

O 5G suporta três tipos de sessão PDU:

- **Sessão PDU do tipo IP:** numa sessão do tipo IP o UE é conectado à uma rede IP externa, como a Internet ou o IMS. O UE é identificado na rede usando um endereço IPv4 e/ou um endereço IPv6, trazendo assim três tipos de sessão: IPv4, IPv6 ou IPv4v6 (COX, 2020);
- **Sessão PDU do tipo *Ethernet*:** nas sessões do tipo *Ethernet* o UE troca pacotes *Ethernet* com uma rede *Ethernet* externa sobre o 5GC. O UE é identificado com um endereço MAC (*Medium Access Control*) de 48 *bits*, assim como acontece tipicamente em redes *Ethernet*. Por conta dos *frames Ethernet* não possuírem informações como os cabeçalhos TCP, UDP e/ou IP, esse tipo de sessão se torna valiosa para comunicações mMTC e URLLC. No entanto, sessões PDU do tipo *Ethernet* ainda estão na fase de implementação por parte do 3GPP, não estando disponíveis para uso geral (ROMMER, 2020);
- **Sessão PDU não-estruturada:** em sessões PDU não-estruturadas o 5GC não tem nenhuma informação a respeito do protocolo usado pela DN. Esse tipo de sessão é limitada pois os dados são tratados como *bits* não estruturados, não sendo possível haver qualquer tipo de classificação ou controle sobre os mesmos. O caso de uso principal para esse tipo de sessão é o de protocolos IoT,

como o 6LoWPAN (*IPv6 over Low-Power Wireless Personal Area Networks*) (ROMMER, 2020).

2.3.3 Tecnologias do 5G

Para preencher todos os requisitos citados acima, o 5G usa tecnologias como virtualização das funções de rede, redes definidas por *software* e fatiamento de rede, de modo a fornecer recursos flexíveis e que possam ser reconfigurados facilmente para fornecer suporte a novos requisitos à medida que esses surgem. Buscando taxa de dados mais altas e robustez no sistema de rádio, a interface aérea suporta frequências mais altas e faz uso de múltiplas antenas, o que somente é possível graças a tecnologias como MIMO e *beamforming*.

Maiores detalhes do funcionamento dessas tecnologias, que são vitais para o desenvolvimento do 5G, podem ser encontrados nas Subseções seguintes.

2.3.3.1 Network Function Virtualization

A infraestrutura de um sistema de telecomunicação móvel é composta por um grande número de elementos, como roteadores, *firewalls*, sistemas de sinalização de canais, etc. Tradicionalmente esses elementos são implementados usando um *hardware* específico, de fornecedores específicos, e que foram concebidos essencialmente para desempenhar uma única tarefa. Essa abordagem traz inúmeras limitações: o *hardware* é proprietário, por isso está vinculado ao fabricante; é fabricado em pequenas quantidades, por isso é caro; e por conta disso o processo de atualização ou aprimoramento da rede à medida em que surgem novos requisitos se torna problemático.

Uma solução atual para esse problema é a NFV (*Network Function Virtualization*), onde o objetivo é implementar os elementos da rede através de *software*, por meio de funções de rede virtualizadas. Essa abordagem traz vantagens pois o *hardware* é genérico, e portanto mais barato, além de que é mais fácil e rápido escrever e/ou atualizar um *software* (se comparado com a atualização de *hardware*) (COX, 2020).

O conceito de NFV, da mesma maneira, pode ser aplicado à RAN. Nesse cenário, além dos benefícios genéricos trazidos pela NFV, uma RAN virtualizada é mais segura, o espaço ocupado pela célula é menor, e o sistema pode ser expandido de forma ágil e barata, visto que essa operação dispensa equipamentos caros e até a necessidade de uma equipe especializada (PENTTINEN, 2019).

2.3.3.2 Software-Defined Networking

Outra tecnologia essencial para o funcionamento do 5G é a SDN (*Software-Defined Networking*), em português, rede definida por *software*. A SDN se caracteriza como uma forma de conceber a arquitetura de rede, tendo como principal diferença em relação às

redes tradicionais o controle por meio de uma arquitetura definida em nível de *software* e baseada em API (PENTTINEN, 2019).

A característica principal de uma rede definida por *software* é a separação dos planos de controle e de dados, de modo que as funções de rede sejam centralizadas.

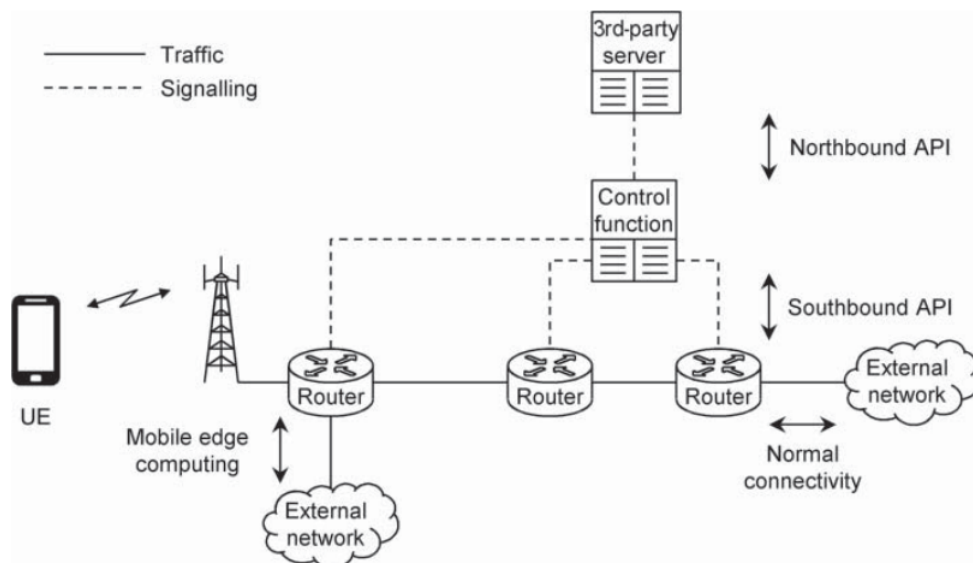
Uma vez centralizadas, essas funções podem se comunicar com aplicações terceiras através de uma API chamada *northbound*. Isso permite que essas aplicações efetuem o controle e monitoramento das funções de rede sem se preocupar com os pormenores da comunicação (COX, 2020).

As aplicações terceiras não conhecem nenhum detalhe da implementação física da rede. Estes estão confinados à API *southbound*, por meio da qual as funções do plano de controle configuram as do plano de dados (COX, 2020).

É importante salientar as diferenças entre NFV e SDN: a primeira separa o *software* do *hardware* ao virtualizar recursos da rede, enquanto a segunda separa o plano de controle do plano de dados. Por isso, as duas são complementares, e não dependentes. Isso significa que uma rede pode possuir ambas, ou somente uma delas.

A Figura 9 mostra a arquitetura de uma rede que faz uso dessas duas tecnologias.

Figura 9 – *Software-Defined Networking*.



Fonte: (COX, 2020).

2.3.3.3 Network Slicing

De maneira simplificada, o *Network Slicing* pode ser definido como uma configuração de rede que permite que múltiplas redes virtualizadas e independentes sejam criadas acima de uma infraestrutura física e compartilhada. Isso só é possível por meio das duas

tecnologias já citadas acima, a NFV e a SDN. O *Network Slicing* divide a rede nas chamadas *slices*, ou em português, fatias, que são faixas independentes da rede, configuradas conforme a necessidade do usuário.

O *Network Slicing* é essencial para o modelo de negócios NaaS (*Network as a Service*) do 5G (MAILER, C, Acesso em: 10 nov. 2022).

Cada fatia é otimizada com base no tráfego que ela carrega. Com isso, a fim de atender cada um dos casos de uso do 5G, como o eMBB, mMTC e URLLC, as diferentes fatias são otimizadas, respectivamente, visando o alto tráfego de dados, a alta densidade de equipamentos conectados, e alta confiabilidade junto de baixa latência (COX, 2020). Além disso, pode haver o uso de diferentes fatias para atender um caso de uso específico, como, por exemplo, usar fatias separadas para áudio e vídeo ao atender o eMBB.

Uma consideração importante a ser feita é ressaltar que as fatias são isoladas umas das outras, especialmente no que diz respeito à segurança e ao transporte de dados. Por conta disso, uma aplicação crítica como o URLLC não é afetada em caso de congestionamento de uma fatia da mesma rede que esteja lidando com alto tráfego de Internet, por exemplo.

2.3.3.4 *Beamforming*

Tecnologia essencial para a implementação do 5G, o *beamforming* se trata de uma técnica de rádio que tem como objetivo direcionar a transmissão diretamente para o receptor, ou esperar uma transmissão vinda da direção do emissor. Essa abordagem melhora a razão sinal-interferência-mais-ruído SINR (*Signal-to-Interference-Plus-Noise Ratio*), atingindo-se maior taxa de transmissão (MAILER, C, Acesso em: 10 nov. 2022).

Há ainda técnicas mais avançadas de uso para o *beamforming*, como o *beamforming* adaptativo, em que o arranjo de antenas possui múltiplos feixes dinamicamente controlados que otimizam a conexão para um dispositivo específico.

2.3.3.5 *MIMO*

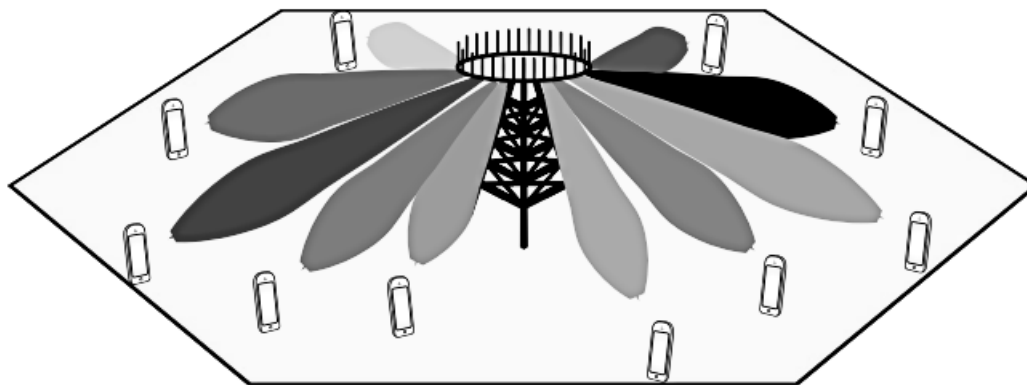
O MIMO se caracteriza por ser uma tecnologia que utiliza múltiplas antenas de transmissão e recepção - e consequentemente canais separados - com diferentes caminhos de propagação, para realizar a transmissão de fluxos de dados entre emissor e receptor numa mesma frequência. O receptor, buscando uma melhor qualidade do sinal recebido, pode combinar ou selecionar o melhor dos diferentes sinais. O termo mMIMO (*massive Multiple Input Multiple Output*) indica que uma quantidade muito maior do que 8 antenas de transmissão e recepção serão utilizadas na estação base (MAILER, C, Acesso em: 10 nov. 2022).

Quando apenas uma estação base com múltiplas antenas se comunica com um único UE, está sendo realizado o SU-MIMO (*Single User Multiple Input Multiple Output*). No MU-MIMO (*Multi User Multiple Input Multiple Output*), a estação base com múltiplas

antenas se comunica com mais de um UE, cada um tendo uma ou múltiplas antenas (XIANG; ZHENG; SHEN, 2017).

Ambos os casos fazem o uso da tecnologia descrita na subseção 2.3.3.4, o *beamforming*. A Figura 10 ilustra o uso do MU-MIMO.

Figura 10 – MU-MIMO e *beamforming*.



Fonte: (XIANG; ZHENG; SHEN, 2017).

2.4 DEFINIÇÕES DA SEXTA GERAÇÃO - 6G

Apesar da implantação e implementação do 5G ainda estarem acontecendo, as discussões sobre a sexta geração da rede móvel, o 6G, já vem sendo feitas em diversos lugares do mundo. O 5G, através dos seus casos de uso, vai atender a maior parte das demandas da população e dos dispositivos que se conectarão à rede móvel: pode-se obter alta taxa de transmissão de dados com o eMBB, baixa latência através do URLLC, e quantidade elevada de dispositivos conectados por meio do mMTC.

No entanto, a limitação do 5G é a capacidade de se atender apenas um requisito chave de cada cenário, o que deixa diversas aplicações listadas pelo IMT-2020 sem suporte (PROJETO BRASIL 6G, 2021a). Como exemplo, pode-se citar a telemedicina, onde os requisitos simultâneos são uma elevada velocidade de comunicação, para suprir a transmissão de dados e imagens, e baixa latência, para a transmissão de comandos e informações de realimentação de movimento (PROJETO BRASIL 6G, 2021a).

Visando cobrir novas demandas da rede, os prováveis casos de uso do 6G foram divididos em seis famílias (PROJETO BRASIL 6G, 2021a):

- **Cobertura Mundial Extrema:** visa atender a demandas por comunicação em qualquer lugar do planeta, inclusive áreas remotas e de difícil acesso;

- **Agropecuária do Futuro:** visa aumentar a produtividade e qualidade de plantações e criações — além da redução de custos e aperfeiçoamento de todo o processo — através da pecuária, agricultura e logística de alta precisão;
- **Gêmeos Digitais em Larga Escala:** visa melhorar o mundo físico com auxílio do mundo virtual, através de uma integração mais próxima entre ambos;
- **Interações Remotas Avançadas:** propõe novas formas de interação e experiências imersivas, como comunicação tátil imersiva, telemedicina imersiva, robôs móveis interativos, entre outros;
- **Zonas Seguras Invisíveis:** através de sensoriamento avançado, pretende resolver problemas como a criminalidade, violência, furtos e acesso a áreas não autorizadas;
- **Comunicação Espacial:** tem como objetivo resolver problemas atuais na comunicação da Terra para a Lua e também na própria superfície da Lua.

Para atender essa futura demanda diversas tecnologias estão sendo propostas. Dentre as principais, pode-se citar: sistemas com múltiplas portadoras não ortogonais, integração de sistemas ópticos e sem fio, uso de frequências nas bandas de sub-THz e THz⁶, múltiplo acesso não ortogonal, entre outras tecnologias disruptivas. Dentre as novas tecnologias, o uso de IA (Inteligência Artificial) vai ter grande destaque na rede móvel de sexta geração (PROJETO BRASIL 6G, 2021b).

2.5 PROTOCOLOS ETHERNET INDUSTRIAL

O termo "*Ethernet* Industrial" se refere a uma série de expansões feitas ao padrão *Ethernet* — definido no IEE 802.3 — com objetivo de adequá-lo ao uso na indústria; a fim de se permitir a comunicação em tempo real entre controladores, máquinas e sensores (SIEMENS, 2008).

Atualmente, são vários os protocolos industriais baseados no *Ethernet*, dentre os quais pode-se citar: EtherCAT, Sercos III, Ethernet/IP, PROFINET⁷, CC-Link IE Field e Modbus TCP (5G ACIA, 2019). Esses protocolos são, portanto, conhecidos como protocolos *Ethernet* Industrial. Em comparação com protocolos usualmente utilizados em redes, estes têm necessidades específicas (SIEMENS, 2008):

- Encaminhamento na camada 2 (sem endereço IP), usualmente feito através do endereço MAC ou VLANs;
- Transmissão de dados determinística, com tempos de resposta e taxas de dados garantidos;

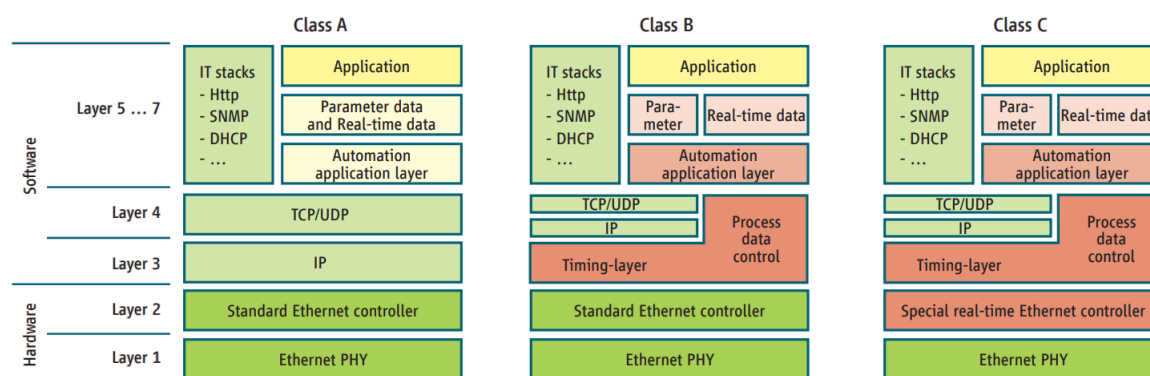
⁶ As ondas THz (*terahertz*) estão na região do espectro eletromagnético entre 300 gigahertz (3×10^{11} Hz) e 3 terahertz (3×10^{12} Hz), sendo o terahertz igual a um trilhão de hertz.

⁷ O PROFINET foi o protocolo escolhido pelo autor para validar a solução proposta neste trabalho, que trata de encapsular e enviar pacotes de protocolos industriais baseados no *Ethernet* através da rede 5G.

- Tempos de ciclo de comunicação curtos e ciclos de comunicação simultâneos com tempo variante;
- Proteção contra falha de componentes;
- Sincronização altamente precisa.

Os protocolos *Ethernet* Industrial foram divididos em três classes, que variam em relação ao desempenho e em parâmetros de implementação, como a necessidade de comunicação em tempo real e o custo. Normalmente, o desempenho da rede melhora da Classe A para a Classe B, e assim sucessivamente. A divisão das classes, bem como sua organização em relação ao modelo OSI, pode ser vista na Figura 11.

Figura 11 – Classificação dos protocolos *Ethernet* Industrial.



Fonte: (5G ACIA, 2019).

A Classe A utiliza *hardware Ethernet* comercial e a pilha TCP/IP para comunicação. Alguns dos protocolos presentes nessa classe são: Ethernet/IP, Modbus TCP e Fieldbus (5G ACIA, 2019).

A Classe B também faz uso de *hardware Ethernet* comercial padrão, mas não emprega a pilha TCP/IP na comunicação. Ao invés disso, ela implementa um protocolo de dados dedicado que é transmitido diretamente dentro do *frame Ethernet*. Protocolos pertencentes a esta classe são: PROFINET RT, POWERLINK e EtherCAT Automation Protocol (EAP) (5G ACIA, 2019).

A Classe C é a mais performática de todas pois emprega *hardware* dedicado até a camada 2. Como exemplo de protocolos pertencentes a esta classe, têm-se: EtherCAT, PROFINET IRT, Sercos III e CC-Link IE (5G ACIA, 2019).

A rede industrial *Ethernet* não é otimizada apenas para as necessidades da comunicação em tempo real, ela também foi pensada para ser de fácil utilização em maquinários. Isso é possível por oferecer uma configuração simples e monitoramento de diagnóstico da rede, a fim de reduzir os tempos de configuração dos equipamentos e o tempo de

inatividade em caso de falhas. Normalmente, as máquinas industriais operam 24 horas por dia, 7 dias por semana, o que requer 100% de disponibilidade de rede.

2.6 OPENAIRINTERFACE

O OAI (OpenAirInterface) é uma plataforma de *software* de código aberto criada com intuito de oferecer ferramentas para o desenvolvimento de redes de telecomunicações móveis definidas por *software* com base nos padrões 3GPP. O projeto é mantido pelo Eureka-147, um instituto francês de pesquisa em telecomunicações (MALLASEN QUINTANA, S., Acesso em: 7 dez. 2022). O OpenAirInterface implementa o NG-RAN e o 5GC em *hardware* genérico, subdivididos em dois projetos (FLORIAN, V. ET AL, Acesso em: 7 dez. 2022).

O OpenAirInterface segue o conceito Open-RAN, que tem a premissa de implementar o rádio do 5G através de iniciativas de código aberto, removendo assim a necessidade de estar preso a tecnologias proprietárias e garantindo interoperabilidade e flexibilidade para escolher diferentes fornecedores durante a implantação de rede.

Usando o simulador OAI, pesquisadores e desenvolvedores podem avaliar o desempenho de protocolos e algoritmos de rede sem fio em várias condições e cenários. Isso permite que eles identifiquem possíveis problemas e otimizem o desempenho de seus sistemas sem fio antes de implantá-los em ambientes do mundo real.

2.7 L2TP

L2TP, em português Protocolo de Tunelamento de Camada 2, é um protocolo de túnel que fornece um método de encapsulamento para *frames* do tipo PPP, permitindo assim a transmissão de dados do tipo L2 através de uma rede IP, ou L3. Foi definido inicialmente na RFC (*Request for Comments*)⁸ 2661 (IETF, Acesso em: 03 nov. 2022[b]), de agosto de 1999, e é baseado em dois protocolos de túnel mais antigos: o L2F (*Layer 2 Forwarding Protocol*) da Cisco, e o PPTP (*Point-to-Point Tunneling Protocol*) da Microsoft. É bastante utilizado na implementação de VPN (*Virtual Private Network*), quase sempre em conjunto com IPsec - já que o mesmo não oferece nenhum tipo de criptografia ou confiabilidade por si só.

Uma nova versão do protocolo, chamada de L2TPv3 (*Layer 2 Tunneling Protocol version 3*), definida na RFC 3931 (IETF, Acesso em: 03 nov. 2022[a]), datada de março de 2005, foi criada com objetivo de fornecer recursos de segurança adicionais, encapsulamento melhorado e a capacidade de conduzir outros enlaces de dados além do PPP, como por exemplo *frames* do tipo *Ethernet*.

⁸ RFC (*Request for Comments*) (em português "pedido de comentários"), são documentos técnicos desenvolvidos por indivíduos e organizações do meio tecnológico e mantidos pela IETF (Internet Engineering Task Force), que têm como objetivo descrever comportamentos, métodos, pesquisas e inovações aplicadas ao funcionamento da Internet e a sistemas conectados à Internet.

O túnel L2TP é estabelecido através do encapsulamento de um *frame* L2TP dentro de um pacote do tipo UDP, que é então encapsulado dentro de um pacote do tipo IP. O endereço de origem e destino desse pacote IP são os responsáveis por definir os *endpoints* da conexão a ser criada. Uma vez que um túnel é estabelecido, o tráfego de dados é bidirecional (KERNEL.ORG, Acesso em: 03 nov. 2022).

O protocolo L2TP separa os *frames* de dados e de controle. Os *frames* de dados são manuseados pelo *driver* do *Kernel*, enquanto os *frames* de controles são manuseados no espaço do usuário. Os *frames* de controle L2TP transportam mensagem entre clientes e servidores e são utilizados para configurar/desmontar túneis e sessões (KERNEL.ORG, Acesso em: 03 nov. 2022).

Cada túnel é implementado utilizando um soquete UDP ou L2TPIP. Cada sessão L2TP cria interfaces de rede que seguem a nomenclatura *l2tpethN*. Para cada túnel e para cada sessão que utilize o túnel são atribuídos, respectivamente, um *tunnel_id* e *session_id* exclusivos. Esses *ids* são usados pelo *driver* L2TP para a tomada de decisões de como lidar com os pacotes (KERNEL.ORG, Acesso em: 03 nov. 2022).

2.8 LINUX NAMESPACES

Namespaces é uma funcionalidade de sistemas operacionais Linux introduzida inicialmente no *Kernel* v2.4.19, em agosto de 2002, através do suporte ao *mount namespace* (KERRISK, Acesso em: 15 nov. 2022), que vêm evoluindo ao longo dos anos (a partir do *Kernel* v5.6 o Linux já oferece suporte a oito tipos diferentes de *namespaces*) e até o momento não teve sua implantação completada. De acordo com o manual do Linux, um *namespace* pode ser definido como:

Um *namespace* envolve um recurso do sistema global em uma abstração que faz parecer aos processos dentro do *namespace* que eles têm sua própria instância isolada do recurso global (LINUX, Acesso em: 15 nov. 2022[a]).

Em outras palavras, um *namespace* é capaz de definir um conjunto de recursos que um processo pode usar. Com isso, pode-se fornecer aos processos sua própria visão do sistema, construindo assim ambientes isolados e independentes uns dos outros (KIROV, Acesso em: 15 nov. 2022[a]). De forma geral, ele permite o particionamento de recursos globais do sistema operacional, como recursos de rede, ferramentas de IPC (*Inter-Process Communication*) ou pontos de montagem.

Um ponto de destaque dos *namespaces* é que eles limitam os recursos disponíveis aos processos sem que os mesmos saibam tais limitações. Além disso, os *namespaces* são um aspecto fundamental para a implementação de *containers*⁹ no sistema Linux.

⁹ Um *container* é um ambiente isolado utilizado para empacotar aplicações, que virtualiza o sistema operacional, e permite que múltiplos sistemas isolados rodem dentro de um outro sistema operacional.

2.8.1 Process ID (PID) namespace

Um PID (*Process IDentification*), é um número de identificação que o Linux utiliza para manter o rastreamento de uma tarefa em funcionamento no sistema. Quando uma aplicação é iniciada, um PID único é gerado e associado à ela. Para que o sistema tenha uma interação correta com suas tarefas, uma aplicação sempre é referenciada pelo seu PID.

Historicamente o *kernel* Linux mantém uma única árvore de processos. A estrutura de dados dessa árvore contém uma referência de cada processo que está em execução numa hierarquia pai-filho. Essa estrutura é mantida no *procfs*, um pseudo *filesystem*¹⁰ que nos permite interagir e observar as estruturas internas do *kernel* Linux em formato de arquivos.

Um processo, satisfazendo certas condições e tendo privilégios suficientes pode inspecionar ou até mesmo encerrar outro processo.

Com a introdução dos *namespaces*, tornou-se possível ter várias árvores de processo aninhadas. Com isso, cada árvore pode ter um conjunto inteiramente isolado de processos. Dessa maneira, processos que pertençam a árvores diferentes não podem inspecionar ou encerrar uns aos outros - na verdade, eles não tem nem mesmo como saber de sua existência (KIROV, Acesso em: 15 nov. 2022[a]).

Toda vez que um sistema Linux é iniciado ele cria um único processo (em sistemas modernos, esse processo geralmente é o *system daemon*). O *systemd* é situado no nó raiz da árvore de processos, e tem o seu PID=1. Esse processo é responsável por iniciar os outros serviços e *daemons* vitais para o sistema, que são então representados como seus filhos na árvore de processos (todos tendo PIDs > 1, e diferentes uns dos outros).

O PID *namespace* permite a um processo a criar sua própria sub-árvore de processos independente, com seu próprio processo com PID=1. O processo que faz isso continua na sua árvore original, mas seu filho (contido no *namespace* recém criado), se vê como o nó raiz de sua própria sub-árvore (KIROV, Acesso em: 15 nov. 2022[a]).

Com a isolação provida pelo PID *namespace*, processos presentes no *namespace* filho não sabem da existência de seu processo pai. O contrário, no entanto, é totalmente possível - visto que o processo pai tem uma visão completa da sub-árvore criada pelo processo filho.

2.8.2 Network (net) namespace

Network namespaces virtualizam a pilha e as funções de rede (LINUX, Acesso em: 15 nov. 2022[b]). Cada *namespace* criado tem suas próprias interfaces, tabela de roteamento, conjunto de endereços IP, regras de *firewall*, lista de portas disponíveis e demais recursos relacionados à rede (EDGE, Acesso em: 15 nov. 2022). Cada interface de

¹⁰ *Filesystem* é um método utilizado pelo sistema operacional para controlar como os dados são armazenados e recuperados. O termo pseudo *filesystem* é válido nesse contexto pois se refere a arquivos que não são "reais", e sim, informações referentes ao sistema em tempo de execução.

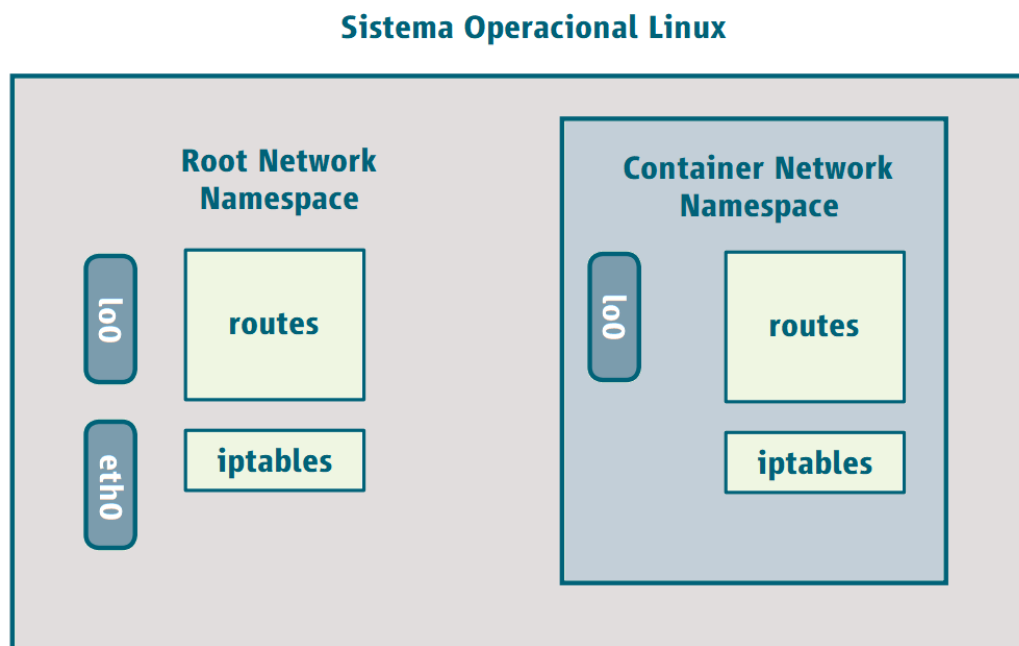
rede do sistema está presente em apenas um *namespace*, mas pode ser movida entre os *namespace* existentes.

Quando criado, um *network namespace* recebe apenas uma interface *loopback* (KIROV, Acesso em: 15 nov. 2022[a])¹¹, que vem desabilitada por padrão:

```
jose@ubuntu:~$ sudo ip netns add netnstest           # creating the namespace
jose@ubuntu:~$ sudo ip netns exec netnstest bash    # entering the namespace
root@ubuntu:/home/jose# ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

O *namespace* não tem acesso nem mesmo à interface de rede física do sistema; em outras palavras: ele não é acessível pela rede (KIROV, Acesso em: 15 nov. 2022[a]). A Figura 12 ilustra esse cenário, com o sistema e o *namespace* sendo "containers" isolados, cada um com suas funções de rede distintas.

Figura 12 – *Network namespace*.



Fonte: O Autor.

Para que um processo dentro do *namespace* seja acessível pela rede, um par de interfaces virtuais¹² precisa ser criado. O *namespace* pai retém uma das interfaces, enquanto

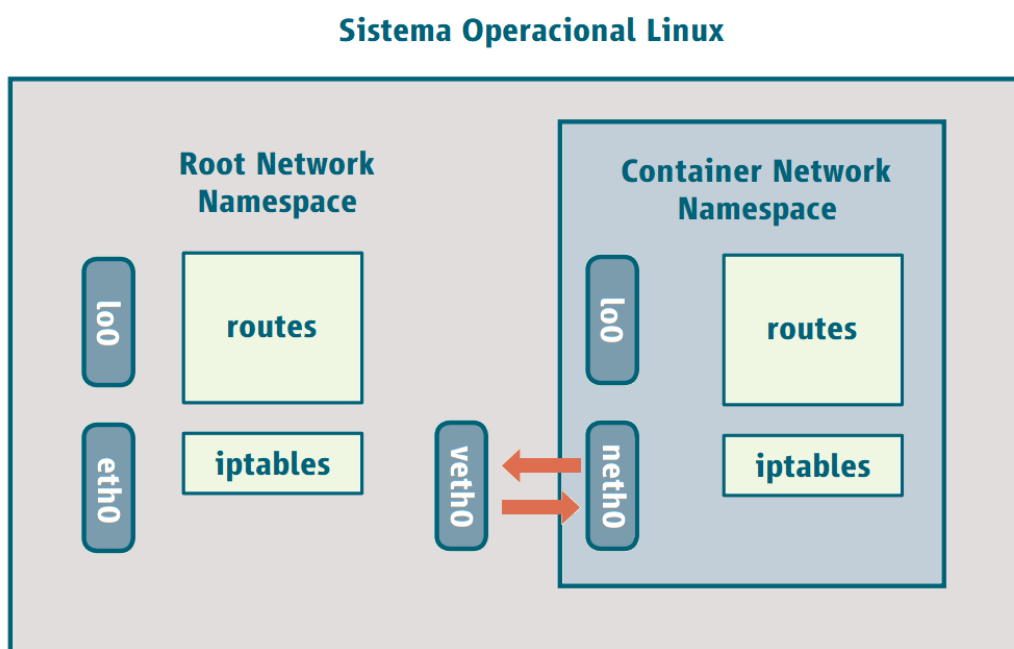
¹¹ A interface *loopback* é um tipo de interface de rede virtual através da qual aplicações podem se comunicar quando estão rodando na mesma máquina. Ela é implementada inteiramente a nível de *software*, não passando pacotes para nenhum controlador de interface de rede.

¹² *veth* é uma interface *Ethernet* virtual que pode ser usada como um túnel entre *namespaces* – a fim de se criar uma bridge com uma interface física em outro *namespace* – ou também como um dispositivo de rede independente.

a outra é movida ao *namespace* filho (KIROV, Acesso em: 15 nov. 2022[a]). Essas duas interfaces vêm com um cabo virtual, ou seja: o que entra em um lado sai no outro. Uma vez que o par é criado, é possível estabelecer *bridges*¹³ ou até mesmo rotear pacotes entre os *namespaces*.

A Figura 13 ilustra o estado dos *namespaces* após a criação do par de interfaces virtuais. O que entra na interface "veth0" sai na interface "neth0", e vice-versa (como se fosse um Linux *pipe*); permitindo, assim, a comunicação entre os *containers*.

Figura 13 – *Network namespace* com as interfaces *veth*.



Fonte: O Autor.

Quando um *network namespace* é destruído, as suas interfaces virtuais também são destruídas, e as interfaces físicas que estavam associadas a ele são movidas novamente ao *namespace* inicial (KIROV, Acesso em: 15 nov. 2022[a]).

2.8.3 Outros namespaces

Como dito anteriormente, até a versão 5.6 do *Kernel*, o Linux já oferece suporte a oito tipos diferentes de *namespaces*, sendo a implementação de novos tipos um tópico de discussão presente em lançamentos futuros. Os *namespaces* suportados atualmente são:

- **Mount (*mnt*) namespace:** é uma poderosa ferramenta para criar *filesystems* por processo, particionando virtualmente o *filesystem* do sistema, desse modo,

¹³ Uma *bridge* é um elemento de rede definido por *software* que opera na camada de enlace do modelo OSI e tem o funcionamento similar a um *switch* de rede, encaminhando pacotes entre as interfaces que estão conectadas à ela.

proporcionando a cada *namespace* uma visão única do sistema de arquivos, quais discos estão montados, onde eles estão montados, além do tipo de montagem (apenas leitura ou leitura/escrita) (KIROV, Acesso em: 15 nov. 2022[b]). Quando criado, o *namespace* filho herda o *mount namespace* do pai, porém, qualquer ponto de montagem criado após não é compartilhado entre os *namespaces*;

- **IPC (Inter-Process Communication) namespace:** garante ao *namespace* isolamento em mecanismos de comunicação de processos, como semáforos, fila de mensagens, segmentos da *shared-memory*, etc (KIROV, Acesso em: 15 nov. 2022[b]). Os processos dentro de um *IPC namespace* não podem ver ou interagir com os recursos IPC do *namespace* pai;
- **UTS (UNIX Time-Sharing System) namespace:** permite que um único sistema pareça ter diferentes nomes de *host* e domínio para diferentes processos (KIROV, Acesso em: 15 nov. 2022[b]);
- **User (usr) namespace:** fornece a virtualização, por processo, os atributos de processo GID e UID (KIROV, Acesso em: 15 nov. 2022[b]). Isso garante, por exemplo, que um processo tenha privilégio *root* dentro do seu *namespace* de usuário;
- **Cgroup (Control Group) namespace:** controla a quantidade de recursos de *hardware* (como CPU, RAM, I\O, espaço de disco) disponíveis a um processo (KIROV, Acesso em: 15 nov. 2022[b]);
- **Time namespace:** permite definir *off-sets*, para cada *namespace*, aos *clocks* de sistema CLOCK_MONOTONIC e CLOCK_BOOTTIME (KIROV, Acesso em: 15 nov. 2022[b]).

3 SISTEMA PROPOSTO

Até aqui ficou evidente a preocupação da ITU e do 3GPP com a indústria, especialmente com o conceito de integração vertical¹, no que tange o desenvolvimento do 5G. Isso fica muito claro com a definição das classes mMTC e URLLC (pensadas para comunicações que precisam de segurança, robustez e/ou baixíssimas latências), e ainda mais com as sessões PDU do tipo *Ethernet*, uma forma pura de enviar pacotes *Ethernet* ou protocolos que implementem o *Ethernet* na camada física, como o PROFINET.

O leitor nesse momento pode se questionar o porquê da necessidade de encapsular pacotes (que é o principal objetivo deste trabalho) se o 5G oferece um tipo de sessão especialmente feita para o envio do padrão *Ethernet*. Isso se dá pelo fato do 3GPP não ter disponibilizado ainda a implementação dessa sessão para uso geral. Os motivos que permeiam essa decisão são principalmente relacionados à maneira que a UPF vai se comportar, pois a mesma terá que tomar decisões de roteamento baseadas nos endereços MAC dos pacotes. Além disso, há questões em aberto relacionadas a como o QoS funcionará, visto que não há endereços IP para aplicar os TFT (*Traffic Flow Template*)².

No artigo de (5G ACIA, 2019) são apresentadas algumas possíveis soluções para resolver o problema do envio de protocolos *Ethernet* através da rede 5G. O presente trabalho visa implementar a sugestão mostrada na Seção 6.3, "*Integration via layer 2 tunneling*", que faz o uso de túneis para enviar pacotes L2³ em uma rede L3. O principal ponto positivo da solução proposta é a utilização de tecnologias já conhecidas e consolidadas para resolver esse problema da rede 5G.

A Figura 14 ilustra o modelo proposto no artigo. Nela, o controlador/master (CM) está ligado a uma infraestrutura L2, que estabelece um túnel L2 associado a uma UPF, e transmite os pacotes até sensores e atuadores (S/A) utilizando a infraestrutura do 5G. As linhas simples em vermelho representam uma conexão direta e física, as linhas duplas representam o túnel, enquanto que as linhas tracejadas querem dizer que o meio de transmissão utilizada é a interface de rádio (ao invés do meio físico).

Para a validação e realização da proposta apresentada acima, fez-se uso das tecnologias citadas ao longo deste trabalho. As simulações foram feitas em um sistema hospedeiro Ubuntu. Para simular a interface de rádio do 5G usou-se o OpenAirInterface, detalhado na Seção 2.6. Os túneis L2 foram criados através do L2TPv3, apresentado anteriormente na Seção 2.7. Tanto os blocos 'C/M' como os 'S/A' foram simulados através de *network namespaces* (Seção 2.8.2), pois estes necessitam de ambientes de rede isolados.

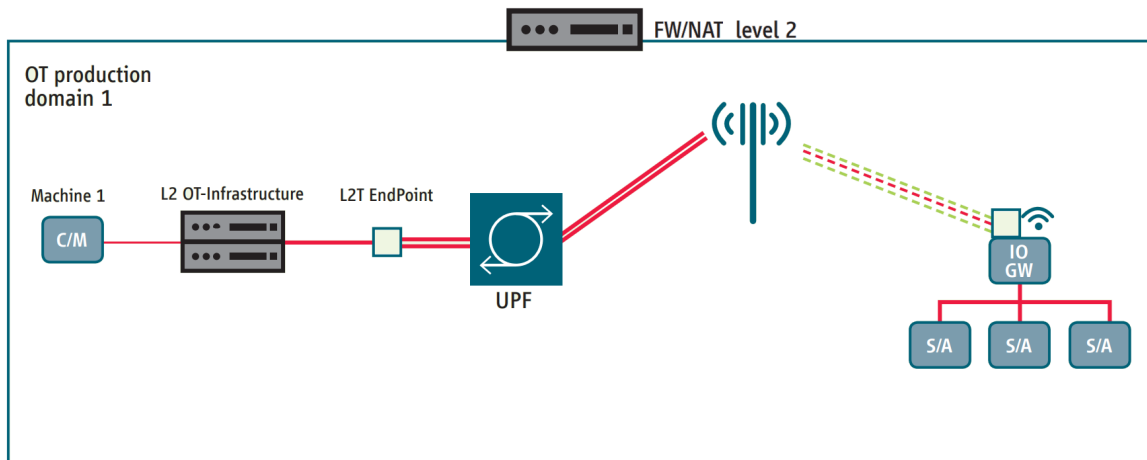
A implementação foi dividida em duas fases e conseqüentemente em duas arquiteturas:

¹ A integração vertical de um sistema de produção está diretamente ligada ao conceito de Indústria 4.0, e tem como principal objetivo permitir que todos os níveis da fábrica estejam conectados, do chão de fábrica até os executivos.

² TFT (*Traffic Flow Template*) é um conjunto de informações usado para mapear um SDF.

³ A notação L_n é usualmente empregada para referenciar as camadas do modelo OSI; onde n é uma das sete camadas.

Figura 14 – Encapsulamento e envio de pacotes no 5G utilizando túneis L2.



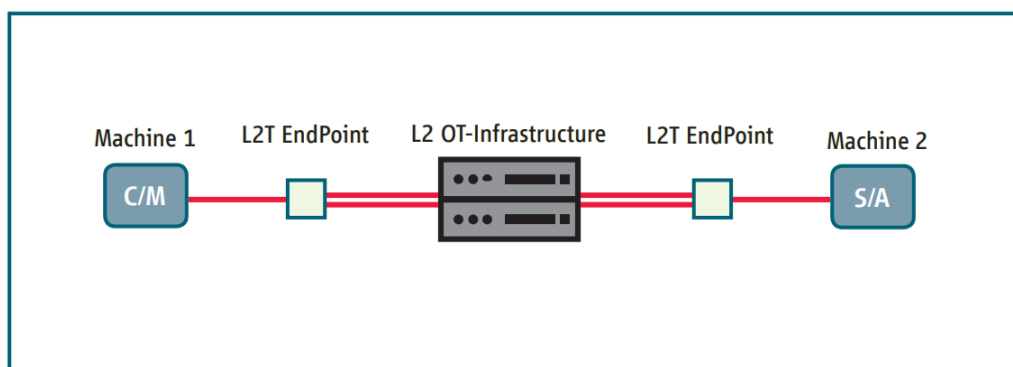
Fonte: (5G ACIA, 2019).

A e B. Ambas serão detalhadas a seguir.

3.1 ARQUITETURA A

Na primeira arquitetura proposta o 5G foi abstraído em sua totalidade; o objetivo aqui é apenas garantir o funcionamento dos túneis. A Figura 15 mostra a topologia proposta.

Figura 15 – Arquitetura A.



Fonte: O Autor (com elementos de (5G ACIA, 2019)).

O bloco "L2 OT-infrastructure"⁴ representa o contexto do sistema hospedeiro. É de responsabilidade dele a criação dos túneis L2. Nesta topologia, os túneis são estabelecidos

⁴ OT (*Operational Technology*) é um acrônimo utilizado na Indústria 4.0 que abrange os elementos relacionados ao maquinário, equipamentos, *softwares*, força de produção e sistemas de comando local das plantas produtivas.

usando o par *veth* de cada *namespace* (as interfaces de rede dos *namespaces* são as mesmas que as da Figura 13). Os túneis estão conectados entre si por meio de uma *bridge*.

O trecho de código⁵ apresentado abaixo é responsável pela criação do ambiente de simulação. O código trata-se de um *Shell script* que cria dois *network namespaces* isolados — que simulam o controlador e os sensores/atuadores — dois túneis L2TPv3 entre o sistema hospedeiro e os *namespaces*, além de uma *bridge* que conecta os dois túneis.

```

1  # create the namespace 'CM' within the network '10.200.1.0/24'
2  create_namespace "CM" "10.200.1.0"
3  # create the namespace 'SA' within the network '10.200.2.0/24'
4  create_namespace "SA" "10.200.2.0"
5
6  # create l2tp tunnels defined as:
7  # <local tun id> <local session id> <local ip> <local udp port>
8  # <remote tun id> <remote session id> <remote ip> <remote udp port>
9  create_l2tp "3000" "1000" "10.200.1.1" "5000" "4000" "2000" "10.200.1.2" "6000"
10 ip link set l2tpeth0 up mtu 1446
11
12 create_l2tp "5000" "3000" "10.200.2.1" "7000" "6000" "4000" "10.200.2.2" "8000"
13 ip link set l2tpeth1 up mtu 1446
14
15 # enter the namespaces and create the same tunnels
16 ip netns exec CM bash -c "${typeset -f}; \
17     create_l2tp 4000 2000 10.200.1.2 6000 3000 1000 10.200.1.1 5000"
18 ip netns exec CM ip link set l2tpeth0 up mtu 1446
19
20 ip netns exec SA bash -c "${typeset -f}; \
21     create_l2tp 6000 4000 10.200.2.2 8000 5000 3000 10.200.2.1 7000"
22 ip netns exec SA ip link set l2tpeth0 up mtu 1446
23
24 # add a bridge between l2tp's interfaces
25 ip link add br0 type bridge
26 ip link set l2tpeth0 master br0
27 ip link set l2tpeth1 master br0
28 ip link set br0 up

```

O resultado esperado para essa arquitetura é que os pacotes enviados em uma ponta do túnel cheguem até a outra ponta. Isso é suficiente, num primeiro momento, para provar que o encapsulamento está sendo feito da maneira correta.

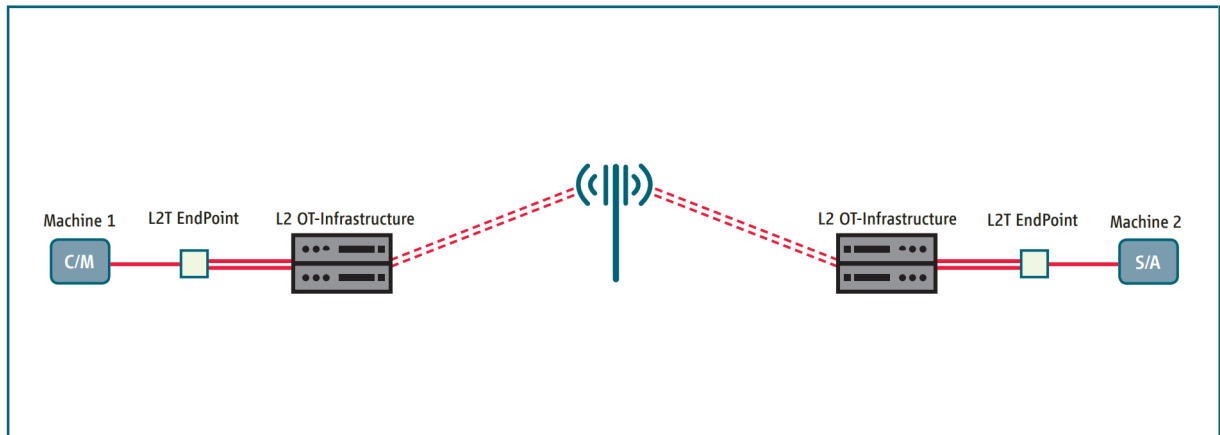
3.2 ARQUITETURA B

A Arquitetura B (mostrada na Figura 16) estende a Arquitetura A ao introduzir o 5G. O túnel L2 que antes era apenas local, agora enviará pacotes por meio da interface

⁵ O código completo é encontrado no Apêndice A. Além disso, as funções usadas foram criadas pelo Autor e estão definidas no Apêndice C.

de rádio do 5G. Nessa arquitetura tanto o controlador quanto os sensores e atuadores são UEs, enquanto a estação base é uma gNB — todos esses elementos são simulados pelo OpenAirInterface (em seus respectivos *namespaces*).

Figura 16 – Arquitetura B.



Fonte: O Autor (com elementos de (5G ACIA, 2019)).

A topologia apresentada acima visava simular um caso de uso real da tecnologia proposta. Tem-se um ambiente simulado análogo a situações encontradas na indústria, onde sensores e atuadores coletam dados e os enviam para o controlador através do túnel, que utiliza a rede do 5G. No entanto, descobriu-se durante a fase de desenvolvimento deste projeto que a aplicação de simulação OAI está limitada a um único UE; o que ocasionou a simplificação da arquitetura proposta.

A nova arquitetura, com apenas um UE, pode ser vista na Figura 17. Nela, um túnel L2 é estabelecido com a gNB. Por mais que a arquitetura seja simplificada, ainda assim a simulação é válida: se os pacotes chegarem na estação base, significa que eles foram corretamente encapsulados e enviados através da interface de rádio do 5G.

O trecho de código apresentado a seguir é responsável pela criação do ambiente de simulação da Arquitetura B (o código completo está presente no Apêndice B).

Aqui, como na Arquitetura A, o controlador é simulado através de um *network namespace*. No sistema hospedeiro é simulado o UE e também são criados os túneis L2: um com o controlador e outro com o gNB (que está sendo simulado em um *network namespace* separado e individual). Por fim, os dois túneis são conectados por meio de uma *bridge*.

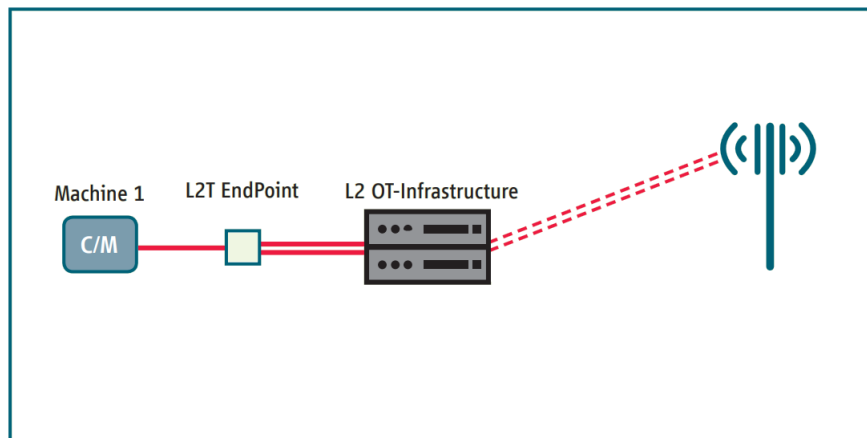
```

1 # create the namespaces for 'gNB' and 'CM'
2 create_namespace "gNB" "10.200.1.0"
3 create_namespace "CM" "10.200.2.0"

```

Após a criação dos *namespaces*, é possível iniciar o OAI. No momento em que este trabalho estava sendo feito, o OAI 5G ainda estava nos estágios iniciais do seu

Figura 17 – Arquitetura B simplificada.



Fonte: O Autor (com elementos de (5G ACIA, 2019)).

desenvolvimento. Por conta disso, a simulação é realizada utilizando os parâmetros *phy-test* e *noS1*, que resulta apenas na criação do NG-RAN. Isso significa que não há nenhuma função do *core* envolvida; há apenas uma conexão entre gNB e UE, que é simulada por meio de interfaces TUN⁶, e que carrega apenas tráfego do plano de dados.

```
# run uE on host
eth5g@eth5g-VirtualBox:~/openairinterface5g/build$ sudo ./nr-uesoftmodem \
-d --rrc_config_path . --nokrnmod --phy-test --rfsim \
--rfsimulator.serveraddr 10.200.1.2 --noS1

# run gNB inside its namespace
eth5g@eth5g-VirtualBox:~/openairinterface5g/build$ sudo ip netns exec gNB bash
root@eth5g-VirtualBox:~/openairinterface5g/build# ./nr-softmodem \
-0 ../../ci-scripts/conf_files/gnb.band78.tm1.106PRB.usrpn300.conf \
--parallel-config PARALLEL_SINGLE_THREAD --rfsim --phy-test \
--rfsimulator.serveraddr server --noS1 -d
```

Com UE e gNB sendo simulados, pode-se estabelecer um dos túneis L2 utilizando as interfaces TUN recém criadas.

```
4 # create l2tp between host and gNB using OAI interfaces
5 create_l2tp "3000" "1000" "10.200.1.1" "5000" "4000" "2000" "10.200.1.2 6000"
6 ip link set l2tpeth0 up mtu 1446
7
8 # create l2tp between host and CM
9 create_l2tp "5000" "3000" "10.200.2.1" "7000" "6000" "4000" "10.200.2.2" "8000"
10 ip link set l2tpeth1 up mtu 1446
```

⁶ TUN, no Linux, se trata de um dispositivo de rede virtual que simula uma interface de rede L3.

```

11
12 # enter the namespaces and create the same tunnels
13 ip netns exec gNB bash -c "$(typeset -f); \
14     create_l2tp 4000 2000 10.0.1.2 6000 3000 1000 10.0.1.1 5000"
15 ip netns exec gNB ip link set l2tpeth0 up mtu 1446
16
17 ip netns exec CM bash -c "$(typeset -f); \
18     create_l2tp 6000 4000 10.200.2.2 8000 5000 3000 10.200.2.1 7000"
19 ip netns exec CM ip link set l2tpeth0 up mtu 1446
20
21 # add a bridge between the tunnels
22 ip link add br0 type bridge
23 ip link set l2tpeth0 master br0
24 ip link set l2tpeth1 master br0
25 ip link set br0 up

```

O resultado esperado para essa arquitetura é receber, na gNB, um pacote *Ethernet* que foi enviado a partir do UE (neste caso representado pelo CM).

3.3 METODOLOGIA DOS TESTES

Uma vez que os ambientes de simulação foram criados, o passo seguinte é testar o funcionamento dos túneis para garantir que o encapsulamento está sendo feito da maneira correta. A metodologia para isso é bem simples: envia-se pacotes em uma ponta do túnel e espera-se receber o mesmo pacote na outra ponta. O envio dos pacotes ficou a cargo da ferramenta Scapy⁷, utilizada para construir um pacote PROFINET (genérico e com dados fictícios) e enviá-lo a partir da interface do túnel L2, como visto no trecho de código abaixo.

```

Welcome to Scapy (2.4.5)
>>> load_contrib('pnio')
>>> p=PNIORealTimeCyclicPDU(cycleCounter=1024, data=[
...     PNIORealTime_IOxS(),
...     PNIORealTimeCyclicPDU.build_fixed_len_raw_type(4)(data = b'AAA') \
...     / PNIORealTime_IOxS(),
...     PROFIsafe.build_PROFIsafe_class(PROFIsafeControl, 4)(data = b'AAA', \
...     control=0x20,crc=0x424242)/PNIORealTime_IOxS(),
... ])
>>> e=Ether(src='2a:24:af:24:34:bd', dst='0a:d0:7f:7f:98:d6') \
...     / ProfinetIO(frameID="RT_CLASS_1") / p
>>> conf.contribs["PNIO_RTC"].update({'2a:24:af:24:34:bd', '0a:d0:7f:7f:98:d6', \
... 0x8000): [
...     PNIORealTime_IOxS,

```

⁷ Scapy é uma ferramenta de manipulação de pacotes capaz de construir e decodificar pacotes de um grande número de protocolos.

```

...     PNIORealTimeCyclicPDU.build_fixed_len_raw_type(4),
...     PNIORealTime_IOxS,
...     PROFIsafe.build_PROFIsafe_class(PROFIsafeControl, 4),
...     PNIORealTime_IOxS,
... ]})
>>> sendp(e, iface="l2tpeth0") # sends a PROFINET packet using l2tpeth0 interface
Sent 1 packets.

```

O encapsulamento e envio correto dos pacotes por si só já valida e cumpre com o objetivo principal deste trabalho. No entanto, para que a técnica desenvolvida se adeque à classe URLLC, e conseqüentemente possa ser usada no ambiente industrial, a latência da comunicação não pode exceder 1 milissegundo.

Em outras palavras: o tempo entre envio, encapsulamento e recebimento deve ser sempre menor que 1 ms. Portanto, mensurar a quantidade de tempo que o pacote leva para ser encapsulado é de extrema importância.

O autor deste trabalho desenvolveu uma técnica simples mas eficiente para estimar o tempo que os pacotes levam para serem encapsulados: calcula-se o tempo de envio médio de pacotes que não foram encapsulados (ou seja, não foram enviados através do túnel) e compara-se com o tempo de envio médio de pacotes que foram encapsulados. Ao calcular-se a diferença de tempo (e considerando que o caminho percorrido pelos dois pacotes é o mesmo) é possível estimar com precisão o tempo de encapsulamento.

Para isso, criou-se um *script* simples na linguagem *Python* responsável por enviar várias amostras de pacotes com tamanhos variados. Aqui é importante ressaltar que o tamanho máximo do pacote deve respeitar o MTU (*Maximum Transmission Unit*)⁸ do túnel, que foi definido no momento de sua criação.

Cada pacote carrega consigo um *timestamp*⁹ que será adicionado ao seu *payload* no momento do seu envio (isso vai ser importante para calcular o tempo de encapsulamento no momento em que o pacote for recebido).

```

1  import scapy.all as scapy
2  import time
3
4  # packets = MAX_PADDING / PADDING_INTERVAL * SAMPLES
5  MAX_PADDING = 1400
6  PADDING_INTERVAL = 50
7  NUM_SAMPLES = 500
8
9  socket = scapy.conf.L2socket(iface="l2tpeth0")
10

```

⁸ O MTU (*Maximum Transmission Unit*) é o quadro ou pacote de maior tamanho — em *bytes* ou octetos — que pode ser transmitido através de um *link* de dados.

⁹ O *timestamp* adicionado segue o formato *Unix*, que é definido por um valor numérico que representa a contagem de quantos segundos se passaram desde o dia 1 de janeiro de 1970 às 00:00:00 do UTC (*Universal Time Coordinated*).


```

11 for s in range(1, MAX_PADDING+1, PADDING_INTERVAL):
12     for i in range(NUM_SAMPLES):
13         socket.send(scapy.Ether(src="6a:52:64:b5:60:21", dst="2a:5a:be:c4:4b:95", \
14             type=0x8892) / f"{scapy.RandString(size=s)}|{time.time()}")

```

No total são enviados 14.000 pacotes. O primeiro pacote tem um tamanho de 14 *bytes*, pois ele carrega o *timestamp* e 1 *byte* de *padding*¹⁰. O próximo tamanho de pacote terá um incremento de 50 *bytes* em relação ao tamanho anterior, definido pela constante 'PADDING_INTERVAL'. Para cada tamanho de pacote são enviadas 500 amostras, definidas no código pela constante 'NUM_SAMPLES'. Os incrementos de tamanho acontecem até o valor definido pela constante 'MAX_PADDING', que respeita o valor do MTU.

Criou-se também um *script* responsável por receber os pacotes na outra ponta do túnel.

```

1 import scapy.all as scapy
2
3 # packets = NUM_SAMPLE * NUM_INTERVALS
4 NUM_SAMPLE = 500
5 NUM_INTERVALS = 28 # (MAX_PADDING / PADDING_INTERVAL)
6 capture = scapy.sniff(filter="ether src 6a:52:64:b5:60:21", \
7     iface="l2tpeh0", count=NUM_INTERVALS*NUM_SAMPLE)
8
9 time = []
10 size = []
11 for i in range(NUM_INTERVALS):
12     time_sum = 0
13     size_sum = 0
14     for j in range(NUM_SAMPLE):
15         packet = capture[(i*NUM_SAMPLE)+j]
16         time_send = bytes(packet.payload).decode('UTF-8').split('|')[1]
17         time_arrival = packet.time
18         time_diff = (float(time_arrival)-float(time_send))*1000
19         time_sum = time_sum + time_diff
20         size_sum = size_sum + len(packet)
21
22     time.append(time_sum/NUM_SAMPLE)
23     size.append(size_sum/NUM_SAMPLE)

```

Após o recebimento dos pacotes, é possível calcular o tempo de transporte ao comparar-se o *timestamp* presente no *payload* (registrado no momento de envio) com o *timestamp* de recebimento, que é registrado pelo *Kernel*.

Primeiramente os pacotes são capturados através da função '*sniff*' (linha 6). Na linha 16 obtém-se o *timestamp* de envio. Já na linha 17 é obtido o *timestamp* de recebimento.

¹⁰ O *padding* está sendo feito pelo método *RandString*, que gera uma *string* aleatória de tamanho *s*.

Ao calcular-se a diferença de tempo entre envio e recebimento, obtém-se o tempo de transporte.

Por fim, ao comparar-se o tempo de transporte de pacotes que foram encapsulados com pacotes que não foram encapsulados, obtém-se o tempo de encapsulamento.

4 RESULTADOS

Após a execução dos *scripts* de simulação no sistema hospedeiro, tem-se como resultado as duas arquiteturas apresentadas anteriormente. Levando como exemplo a execução do *script* da Arquitetura A, isso é observado a partir dos *namespaces* criados:

```
eth5g@eth5g-VirtualBox:~/tcc/eth_5G$ sudo ip netns
SA (id: 2)
CM (id: 1)
```

dos túneis estabelecidos:

```
eth5g@eth5g-VirtualBox:~/tcc/eth_5G$ sudo ip l2tp show tunnel
Tunnel 5000, encap UDP
  From 10.200.2.1 to 10.200.2.2
  Peer tunnel 6000
  UDP source / dest ports: 7000/8000
Tunnel 3000, encap UDP
  From 10.200.1.1 to 10.200.1.2
  Peer tunnel 4000
  UDP source / dest ports: 5000/6000
```

e das interfaces de rede configuradas:

```
eth5g@eth5g-VirtualBox:~/tcc/eth_5G$ ifconfig
...
br0      Link encap:Ethernet  HWaddr 1e:6f:0c:bf:2c:60
        inet6 addr: fe80::1c6f:cff:febf:2c60/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1446  Metric:1
        RX packets:13 errors:0 dropped:0 overruns:0 frame:0
        TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:860 (860.0 B)  TX bytes:2867 (2.8 KB)
l2tpeth0 Link encap:Ethernet  HWaddr 1e:6f:0c:bf:2c:60
        UP BROADCAST RUNNING MULTICAST  MTU:1446  Metric:1
        RX packets:8 errors:0 dropped:0 overruns:0 frame:0
        TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:752 (752.0 B)  TX bytes:3523 (3.5 KB)
l2tpeth1 Link encap:Ethernet  HWaddr a6:c0:1a:7a:1f:8a
        UP BROADCAST RUNNING MULTICAST  MTU:1446  Metric:1
        RX packets:8 errors:0 dropped:0 overruns:0 frame:0
        TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:752 (752.0 B)  TX bytes:3433 (3.4 KB)
vloc-CM  Link encap:Ethernet  HWaddr 92:69:ed:5a:a5:28
        inet addr:10.200.1.1  Bcast:0.0.0.0  Mask:255.255.255.0
```

```

inet6 addr: fe80::9069:edff:fe5a:a528/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:20 errors:0 dropped:0 overruns:0 frame:0
TX packets:75 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2042 (2.0 KB) TX bytes:10653 (10.6 KB)
vloc-SA Link encap:Ethernet HWaddr 2e:14:fd:3e:95:89
inet addr:10.200.2.1 Bcast:0.0.0.0 Mask:255.255.255.0
inet6 addr: fe80::2c14:fdff:fe3e:9589/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:19 errors:0 dropped:0 overruns:0 frame:0
TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2000 (2.0 KB) TX bytes:10391 (10.3 KB)

```

O ambiente criado para a simulação da Arquitetura B é muito semelhante a nível de sistema operacional e recursos — se comparados os túneis e *namespaces* — com a Arquitetura A. A única diferença notável se dá pelas interfaces TUN do OAI, que têm como função simular o acesso de rádio entre UE e gNB; foi a partir delas que o túnel da Arquitetura B, responsável por enviar os dados através do 5G, foi estabelecido.

```

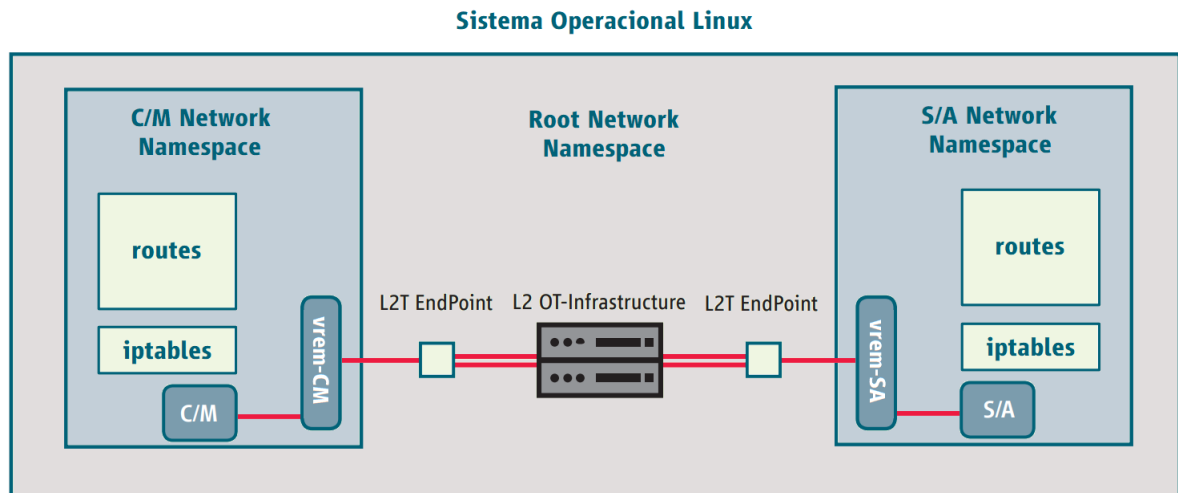
eth5g@eth5g-VirtualBox:~/openairinterface5g/build$ ifconfig
...
oaitun_ue1 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:10.0.1.2 P-t-P:10.0.1.2 Mask:255.255.255.0
inet6 addr: fe80::3e4e:5d2f:e51a:6810/64 Scope:Link
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:1 errors:0 dropped:0 overruns:0 frame:0
TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:48 (48.0 B) TX bytes:96 (96.0 B)
eth5g@eth5g-VirtualBox:~/openairinterface5g/build$ sudo ip netns exec gNB bash
root@eth5g-VirtualBox:~/openairinterface5g/build# ifconfig
...
oaitun_enb1 Link encap:UNSPEC HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
inet addr:10.0.1.1 P-t-P:10.0.1.1 Mask:255.255.255.0
inet6 addr: fe80::e62f:f9a0:992e:f864/64 Scope:Link
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:3 errors:0 dropped:0 overruns:0 frame:0
TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:144 (144.0 B) TX bytes:144 (144.0 B)

```

Nas Figuras 18 e 19 pode-se ver de forma completa os cenários de simulação que foram criados e que serão utilizados para validar a solução proposta. A ilustração foi feita de forma didática e não é totalmente fiel em relação aos recursos vistos pelo sistema

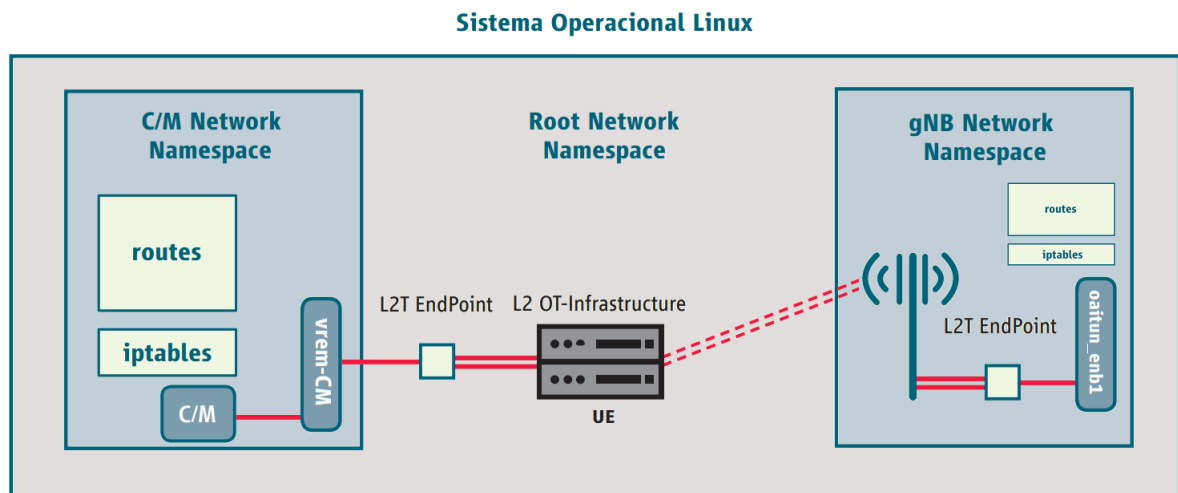
operacional pois mistura elementos "reais"(como as interfaces de rede) com elementos "fictícios"(como 'C/M' e 'S/A').

Figura 18 – Arquitetura A detalhada.



Fonte: O Autor (com elementos de (5G ACIA, 2019)).

Figura 19 – Arquitetura B detalhada.



Fonte: O Autor (com elementos de (5G ACIA, 2019)).

Com os ambientes de simulação já criados, foi possível testar o funcionamento dos túneis. No cenário da Arquitetura A, envia-se os pacotes a partir do *namespace* 'CM', e monitora-se o tráfego de rede na outra ponta do túnel (dentro do *namespace* 'SA'), por meio da ferramenta tcpdump¹.

¹ tcpdump é uma ferramenta do Linux usada para detecção e análise de pacotes de rede.

```
eth5g@eth5g-VirtualBox:~$ sudo ip netns exec SA bash
root@eth5g-VirtualBox:~# tcpdump -tt -v -elni l2tpeth0
tcpdump: listening on l2tpeth0, link-type EN10MB (Ethernet), capture size 262144 bytes
1670719940.951911 2a:24:af:24:34:bd > 0a:d0:7f:7f:98:d6, ethertype (0x8892), length 35:
    0x0000:  8000 8041 4141 0080 4141 4100 2042 4242  ...AAA..AAA..BBB
    0x0010:  8004 0035 00                                ...5.
```

Percebe-se que o mesmo pacote PROFINET enviado anteriormente foi recebido com sucesso (preste atenção nos *frames* de dado 'AAA'). Isso valida o funcionamento do túnel L2 e conseqüentemente o encapsulamento dos pacotes.

O próximo passo agora é validar a Arquitetura B, que estende o túnel à interface de rede do 5G. Para isso, envia-se o mesmo pacote PROFINET a partir do *namespace* 'CM'.

Utilizando novamente o *tcpdump* na ponta do túnel L2, que desta vez está no *namespace* 'gNB', obtém-se o resultado esperado: um pacote PROFINET encapsulado em um túnel de camada 2 e enviado através da interface de rádio do 5G.

```
eth5g@eth5g-VirtualBox:~$ sudo ip netns exec gNB bash
root@eth5g-VirtualBox:~# tcpdump -tt -v -elni l2tpeth0
tcpdump: listening on l2tpeth0, link-type EN10MB (Ethernet), capture size 262144 bytes
1670723398.852674 6a:52:64:b5:60:21 > 2a:5a:be:c4:4b:95, ethertype (0x8892), length 35:
    0x0000:  8000 8041 4141 0080 4141 4100 2042 4242  ...AAA..AAA..BBB
    0x0010:  8004 0035 00                                ...5.
```

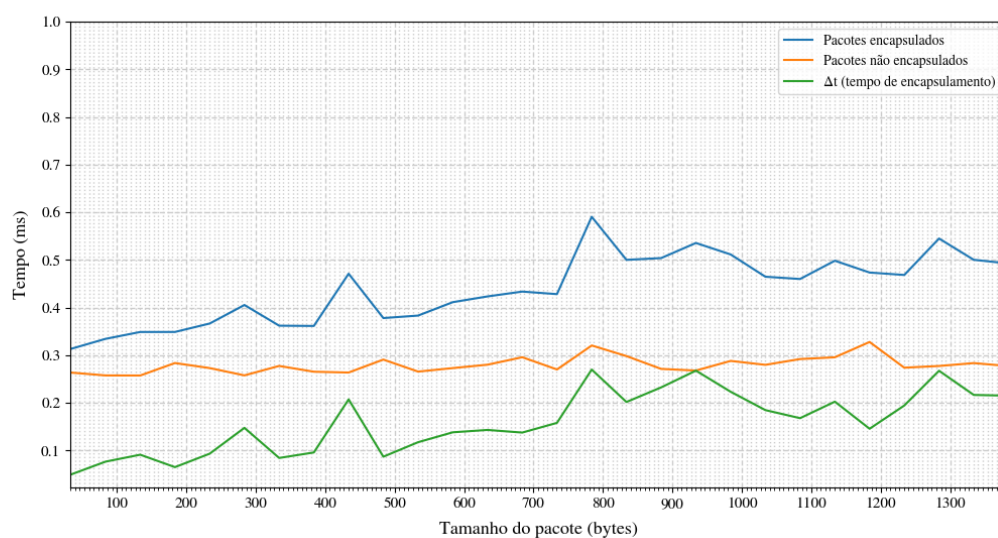
Com os túneis funcionando, foi possível validar se a solução proposta se adequa à classe de uso URLLC, ou seja, se o tempo de encapsulamento não será prejudicial à latência da comunicação. Isso foi feito através da utilização dos *scripts* de envio e captura de pacotes nas pontas do túneis.

A Figura 20 mostra um gráfico construído a partir dos dados coletados pelo *script* responsável por capturar os pacotes. Nela, tem-se o tempo médio de transporte de pacotes encapsulados e não encapsulados em função do tamanho do pacote — bem como a diferença de tempo entre os dois, que pode ser entendida como o tempo de encapsulamento — num gráfico de linhas.

Percebe-se que o resultado foi satisfatório, com o tempo de encapsulamento a todo momento permanecendo bem abaixo de 1 milissegundo. Levando em conta que a latência é definida pelo RTT (*Round-Trip Time*), ou seja, o tempo entre envio, recebimento e resposta, o valor total consumido pelo encapsulamento seria dobrado (pois o pacote é encapsulado no envio da requisição e no envio da resposta); valor que ainda assim fica dentro do limite definido e esperado para a classe URLLC.

Houve também pequenas variações no tempo de encapsulamento dos pacotes que podem ser percebidas ao se fazer a análise do gráfico. A explicação para isso é que todo o ambiente está sendo simulado por *software* — além de ser executado em apenas um

Figura 20 – Gráfico do tempo de encapsulamento em função do tamanho do pacote.



Fonte: O Autor.

computador —, ficando refém de recursos computacionais e sendo afetado por parâmetros como o escalonamento de tarefas por parte do sistema operacional.

5 CONCLUSÕES

A técnica de encapsulamento utilizada neste trabalho, ao fazer uso de ferramentas já conhecidas e bastante utilizadas no ramo de redes de comunicação, cumpre com os objetivos propostos neste trabalho e se mostra como sendo uma solução válida para o problema de enviar protocolos industriais baseados em *Ethernet* através do 5G.

O tempo de encapsulamento do pacote permaneceu em valores aceitáveis até mesmo para aplicações mais críticas presentes na classe URLLC. As pequenas variações no tempo de encapsulamento encontradas após os testes eram de certo modo esperadas e são causadas pelo atraso nas operações desempenhadas pelo sistema operacional, visto que todos os elementos de simulação são definidos por *software*.

O processo de encapsulamento se mostrou também não ser tão variante em relação ao tamanho do pacote; o que faz sentido, visto que o encapsulamento se dá apenas ao adicionar cabeçalhos referentes à transmissão ao próprio pacote.

Contudo, o método desenvolvido não é aconselhável para todos os casos de uso da indústria. Ao adicionar-se ao pacote transmitido informações necessárias para o seu encapsulamento (como um endereço IPv6 de destino, que tem um tamanho de 128 *bits*), bastante informação além do *payload* do pacote é enviada. Isso pode ser um empecilho em alguns casos específicos onde se pode transmitir apenas alguns *bits* — por se ter um plano de dados limitado — por exemplo. A esses, torna-se necessário buscar novos meios. Uma possível solução seria realizar a implementação própria de sessões PDU do tipo *Ethernet*, ou até mesmo do tipo não estruturada.

REFERÊNCIAS

- 3GPP. **3GPP TR 21.915 version 15.0.0 Release 15**. [S.l.: s.n.], Acesso em: 20 nov. 2022. Disponível em:<https://www.3gpp.org/release-15>.
- 3GPP, ETSI. **ETSI TS 138 104**. [S.l.: s.n.], Acesso em: 10 nov. 2022. Disponível em:https://www.etsi.org/deliver/etsi_ts/138100_138199/138104/16.06.00_60/ts_138104v160600p.pdf.
- 5G ACIA. **Integration of Industrial Ethernet Networks with 5G Networks**, 2019.
- 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. 6, 9, 34.
- COX, C. **An Introduction to 5G: The New Radio, 5G Network and Beyond**. [S.l.]: Wiley, 2020. P. 1, 2, 3, 4, 5, 7, 8, 15, 17, 18, 31, 32, 33, 34, 35, 37, 58, 59, 76, 152, 153, 155, 279, 280, 281. ISBN 9781119602682.
- DAVANÇO SILVA, G. H. **Classificação de tráfego por classes de serviço no núcleo 5G**. [S.l.: s.n.], Acesso em: 15 nov. 2022. Disponível em:<https://repositorio.ufsc.br/bitstream/handle/123456789/237577/TCC.pdf>.
- DUSSUAX, G. et al. **Tecnologias de Rede em Telefonia Móvel**. [S.l.: s.n.], Acesso em: 26 out. 2022. Disponível em:https://www.gta.ufrj.br/grad/10_1/movel/index.html.
- EDGE, J. **Namespaces in operation, part 7: Network namespaces**. [S.l.: s.n.], Acesso em: 15 nov. 2022. <https://lwn.net/Articles/580893/>.
- FLORIAN, V. ET AL. **Field Trial of a 5G Non-Terrestrial Network Using OpenAirInterface**. [S.l.: s.n.], Acesso em: 7 dez. 2022. Disponível em:<https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9775203>.
- IETF. **Layer Two Tunneling Protocol - Version 3 (L2TPv3)**. [S.l.: s.n.], Acesso em: 03 nov. 2022. <https://www.rfc-editor.org/rfc/rfc3931>.
- IETF. **Layer Two Tunneling Protocol "L2TP"**. [S.l.: s.n.], Acesso em: 03 nov. 2022. <https://www.rfc-editor.org/rfc/rfc2661>.

- ITU-R. **Recommendation ITU-R M.2083-0**. [*S.l.: s.n.*], Acesso em: 20 nov. 2022. Disponível em:https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf.
- KERNEL.ORG. **L2TP**. [*S.l.: s.n.*], Acesso em: 03 nov. 2022. <https://www.kernel.org/doc/html/v5.8/networking/l2tp.htm>.
- KERRISK, M. **Namespaces in operation, part 1: namespaces overview**. [*S.l.: s.n.*], Acesso em: 15 nov. 2022. <https://lwn.net/Articles/531114/>.
- KIROV, M. **Digging into Linux namespaces - part 1**. [*S.l.: s.n.*], Acesso em: 15 nov. 2022. <https://blog.quarkslab.com/digging-into-linux-namespaces-part-1.html>.
- KIROV, M. **Digging into Linux namespaces - part 2**. [*S.l.: s.n.*], Acesso em: 15 nov. 2022. <https://blog.quarkslab.com/digging-into-linux-namespaces-part-2.html>.
- LINUX. **namespaces(7) — Linux manual page**. [*S.l.: s.n.*], Acesso em: 15 nov. 2022. Disponível em:<https://man7.org/linux/man-pages/man7/namespaces.7.html>.
- LINUX. **network_namespaces(7) — Linux manual page**. [*S.l.: s.n.*], Acesso em: 15 nov. 2022. Disponível em:https://man7.org/linux/man-pages/man7/network_namespaces.7.html.
- MAILER, C. **Plataforma de CORE 5G em nuvem para disponibilização de funções de rede como serviço**. [*S.l.: s.n.*], Acesso em: 10 nov. 2022. P. 21, 22, 23, 24, 25, 27, 28. Disponível em:https://repositorio.ufsc.br/bitstream/handle/123456789/209624/TCC_20201_ChristianMailer.pdf.
- MALLASEN QUINTANA, S. **Deployment and analysis of a 5G NR radio access network based on Open RAN, using USRPs and OpenAirInterface**. [*S.l.: s.n.*], Acesso em: 7 dez. 2022. Disponível em:<http://hdl.handle.net/10251/187834>.
- MICROSOFT. **Private 5G Core**. [*S.l.: s.n.*], Acesso em: 20 nov. 2022. Disponível em:<https://learn.microsoft.com/pt-br/azure/private-5g-core/policy-control>.
- MISHRA, A. R. **Advanced Cellular Network Planning and Optimisation: 2G/2.5G/3G ... evolution to 4G**. [*S.l.*]: John Wiley, 2007. P. 1, 10, 11.

MISHRA, A. R. **Fundamentals of Network Planning and optimisation for wireless communications: 2G, 3G and systems beyond**. [S.l.]: Wiley, 2004. P. 4, 5, 6, 7.

NOLL BARRETO, A. et al. **5G Standalone ou “5G Puro”**. [S.l.: s.n.], Acesso em: 10 nov. 2022. Disponível em:<https://5gdescomplicado.com/2021/05/08/5g-standalone-ou-5g-puro/>.

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

PEREIRA MARTINS, A.; OLIVEIRA, F. C. DE. **A evolução do LTE para o LTE-Advanced Pro**. [S.l.: s.n.], Acesso em: 15 nov. 2022. Disponível em:<https://www.inatel.br/biblioteca/todo-docman/pos-seminarios/seminario-de-redes-e-sistemas-de-telecomunicacoes/2018-1/9944-a-evolucao-do-lte-para-o-lte-advanced-pro/file>.

PROJETO BRASIL 6G. **Casos de Uso e Requisitos para as Redes 6G**, 2021.

PROJETO BRASIL 6G. **Contribuições para a Camada Física de Redes 6G**, 2021.

ROMMER, S. et al. **5G Core Networks: Powering Digitalization**. London: Academic Press, 2020. P. 1, 2, 3, 4, 16, 17, 18, 19, 20, 21, 22, 23, 52, 53, 54, 66, 67, 68. ISBN 978-0081030097.

SIEMENS. **Industrial Communication - SIMATIC NET Industrial Ethernet Networking Manual**, 2008.

TELECO. **4G: Tecnologias de Celular**. [S.l.: s.n.], Acesso em: 04 nov. 2022. Disponível em:https://www.teleco.com.br/4g_tecnologia.asp.

TSG-T. **Definitions used for the Mobile Station/Terminal**. [S.l.: s.n.], Acesso em: 15 nov. 2022. Disponível em:https://www.3gpp.org/ftp/tsg_t/TSG_T/TSGT_05/Docs/PDFs/TP-99203.PDF.

TUDE, E. **GPRS**. [S.l.: s.n.], Acesso em: 05 nov. 2022. Disponível em:https://www.teleco.com.br/tutoriais/tutorialgprs/pagina_3.asp.

VIEIRA PEIXOTO, Wandenberg. **TELEFONIA 3G**. [S.l.: s.n.], Acesso em: 26 out. 2022. Disponível em:https://www.gta.ufrj.br/grad/04_2/telefonია3g/.

XIANG, Wei; ZHENG, Kan; SHEN, Xuemin (Sherman). **5G Mobile Communications**. [S.l.]: Springer International Publishing, 2017. P. 93.

ZAIDI A.; BALDEMAIR, R. **In the race to 5G, CP-OFDM triumphs!** [S.l.: s.n.], Acesso em: 20 nov. 2022. Disponível em:<https://www.ericsson.com/en/blog/2017/5/in-the-race-to-5g-cp-ofdm-triumphs>.

APÊNDICE A – Criação da Arquitetura A

```

1  #!/bin/bash
2  source "./helpers.sh"
3  ACTION="$1"
4
5  main()
6  {
7      if [ "$ACTION" = "add" ]; then
8          # create the namespace 'CM' (10.200.1.0/24) and 'SA' (10.200.2.0/24)
9          create_namespace "CM" "10.200.1.0"
10         create_namespace "SA" "10.200.2.0"
11
12         # create l2tp tunnels defined as:
13         # <local tun id> <local session id> <local ip> <local udp port>
14         # <remote tun id> <remote session id> <remote ip> <remote udp port>
15         create_l2tp "3000" "1000" "10.200.1.1" "5000" "4000" "2000" "10.200.1.2" "6000"
16         ip link set l2tpeth0 up mtu 1446
17         create_l2tp "5000" "3000" "10.200.2.1" "7000" "6000" "4000" "10.200.2.2" "8000"
18         ip link set l2tpeth1 up mtu 1446
19
20         # enter the namespaces and create the same tunnels
21         ip netns exec CM bash -c "${typeset -f}; \
22             create_l2tp 4000 2000 10.200.1.2 6000 3000 1000 10.200.1.1 5000"
23         ip netns exec CM ip link set l2tpeth0 up mtu 1446
24         ip netns exec SA bash -c "${typeset -f}; \
25             create_l2tp 6000 4000 10.200.2.2 8000 5000 3000 10.200.2.1 7000"
26         ip netns exec SA ip link set l2tpeth0 up mtu 1446
27
28         # add a bridge between l2tp's interfaces
29         ip link add br0 type bridge
30         ip link set l2tpeth0 master br0
31         ip link set l2tpeth1 master br0
32         ip link set br0 up
33     else
34         delete_namespace "CM"
35         delete_namespace "SA"
36         delete_l2tp "3000" "1000"
37         delete_l2tp "5000" "3000"
38         ip link del br0
39     fi
40 }
41
42 if [ "$ACTION" != "add" ] && [ "$ACTION" != "del" ]; then
43     echo "You must provide action \"add\" or \"del\"!"
44     exit 1
45 fi

```

```
46
47  if [ $(id -u) -ne 0 ]; then
48    echo "You are required to run this script with sudo or as root!"
49    exit 1
50  fi
51
52  main
53
54  exit 0
55
```

APÊNDICE B – Criação da Arquitetura B

```

1  #!/bin/bash
2  source "./helpers.sh"
3
4  ACTION="$1"
5
6  main()
7  {
8      if [ "$ACTION" = "add" ]; then
9          # create the namespace 'gNB' within the network '10.200.1.0/24'
10         create_namespace "gNB" "10.200.1.0"
11         # create the namespace 'CM' within the network '10.200.2.0/24'
12         create_namespace "CM" "10.200.2.0"
13
14         # run uE on host
15         # $ sudo ./nr-uesoftmodem -d --rrc_config_path . --nokrrnmod --phy-test \
16         #   --rfsim --rfsimulator.serveraddr 10.200.1.2 --noS1
17
18         # run gNB inside its namespace
19         # $ sudo ./nr-softmodem --parallel-config PARALLEL_SINGLE_THREAD --rfsim \
20         #   -D ../../../../ci-scripts/conf_files/gnb.band78.tm1.106PRB.usrpn300.conf \
21         #   --phy-test --rfsimulator.serveraddr server --noS1 -d
22
23         # create l2tp tunnel using gNB and UE interfaces
24         create_l2tp "3000" "1000" "10.200.1.1" "5000" "4000" "2000" "10.200.1.2" "6000"
25         ip link set l2tpeth0 up mtu 1446
26
27         # create l2tp tunnel using veth's
28         create_l2tp "5000" "3000" "10.200.2.1" "7000" "6000" "4000" "10.200.2.2" "8000"
29         ip link set l2tpeth1 up mtu 1446
30
31         # enter gNB namespace and create the tunnel
32         ip netns exec gNB bash -c "${typeset -f}; \
33             create_l2tp 4000 2000 10.0.1.2 6000 3000 1000 10.0.1.1 5000"
34         ip netns exec gNB ip link set l2tpeth0 up mtu 1446
35
36         # enter CM namespace and create the tunnel
37         ip netns exec CM bash -c "${typeset -f}; \
38             create_l2tp 6000 4000 10.200.2.2 8000 5000 3000 10.200.2.1 7000"
39         ip netns exec CM ip link set l2tpeth0 up mtu 1446
40
41         # add a bridge between l2tp's interfaces
42         ip link add br0 type bridge
43         ip link set l2tpeth0 master br0
44         ip link set l2tpeth1 master br0
45         ip link set br0 up

```

```
46     else
47         delete_namespace "gNB"
48         delete_namespace "CM"
49         delete_l2tp "3000" "1000"
50         delete_l2tp "5000" "3000"
51         ip link del br0
52     fi
53 }
54
55 if [ "$ACTION" != "add" ] && [ "$ACTION" != "del" ]; then
56     echo "You must provide action \"add\" or \"del\"!"
57     exit 1
58 fi
59
60 if [ $(id -u) -ne 0 ]; then
61     echo "You are required to run this script with sudo or as root!"
62     exit 1
63 fi
64
65 main
66
67 exit 0
68
```


APÊNDICE C – *helpers functions*

```

1  #!/bin/bash
2
3  next_ip()
4  {
5      local IP NEXT_IP IP_HEX NEXT_IP_HEX
6      IP=$1
7      IP_HEX=$(printf '%.2X%.2X%.2X%.2X\n' `echo $IP | sed -e 's/\./ /g'`)
8      NEXT_IP_HEX=$(printf "%.8X `echo $(( 0x$IP_HEX + 1 ))`)
9      NEXT_IP=$(printf '%d.%d.%d.%d\n' `echo $NEXT_IP_HEX | sed -r 's/(..)/0x\1 /g'`)
10     echo "$NEXT_IP"
11 }
12
13 create_namespace()
14 {
15     local NAME IPN IPL IPR
16     NAME=$1
17     IPN=$2
18     IPL=$(next_ip "$IPN")
19     IPR=$(next_ip "$IPL")
20
21     ip netns add "$NAME"
22     ip link add "vloc-$NAME" type veth peer name "vrem-$NAME"
23     ip link set "vrem-$NAME" netns "$NAME"
24     ip addr add "$IPL"/24 dev "vloc-$NAME"
25     ip link set "vloc-$NAME" up
26
27     iptables -t nat -A POSTROUTING -s "$IPN"/255.255.255.0 -o enp0s3 -j MASQUERADE
28     iptables -A FORWARD -i enp0s3 -o "vloc-$NAME" -j ACCEPT
29     iptables -A FORWARD -o enp0s3 -i "vloc-$NAME" -j ACCEPT
30
31     ip netns exec "$NAME" ip link set dev lo up
32     ip netns exec "$NAME" ip addr add "$IPR"/24 dev "vrem-$NAME"
33     ip netns exec "$NAME" ip link set "vrem-$NAME" up
34 }
35
36 delete_namespace()
37 {
38     local NAME
39     NAME=$1
40     ip link delete "vloc-$NAME"
41     ip netns delete "$NAME"
42 }
43
44 create_l2tp()
45 {

```

```
46  local TUNL SESSL IPL PORTL TUNR SESSR IPR PORTR
47  TUNL=$1
48  SESSL=$2
49  IPL=$3
50  PORTL=$4
51  TUNR=$5
52  SESSR=$6
53  IPR=$7
54  PORTR=$8
55
56  # create tunnel
57  ip l2tp add tunnel tunnel_id "$TUNL" peer_tunnel_id "$TUNR" \
58     encap udp local "$IPL" remote "$IPR" \
59     udp_sport "$PORTL" udp_dport "$PORTR"
60
61  # create session
62  ip l2tp add session tunnel_id "$TUNL" session_id "$SESSL" \
63     peer_session_id "$SESSR"
64 }
65
66 delete_l2tp()
67 {
68     local TUN SESS
69     TUN=$1
70     SESS=$2
71
72     ip l2tp del session tunnel_id "$TUN" session_id "$SESS"
73     ip l2tp del tunnel tunnel_id "$TUN"
74 }
75
```