UNIVERSIDADE FEDERAL DE GOIÁS (UFG)

INSTITUTO DE INFORMÁTICA (INF)

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PABLLO BORGES CARDOSO

Integração de uma Aplicação de Realidade Aumentada com Sistemas 5G seguindo o padrão 3GPP

Goiânia
2024

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[X] Dissertação [ ] Tese [ ] Outro*:_____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Pabllo Borges Cardoso

3. Título do trabalho

Integração de uma Aplicação de Realidade Aumentada com Sistemas 5G seguindo o padrão 3GPP

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professor do Magistério Superior**, em 19/12/2024, às 20:08, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Pabllo Borges Cardoso**, **Discente**, em 20/12/2024, às 15:30, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5052683** e o código CRC **08ED02A0**.

PABLLO BORGES CARDOSO

Integração de uma Aplicação de Realidade Aumentada com Sistemas 5G seguindo o padrão 3GPP

Dissertação apresentada ao Programa de Pós–Graduação em Ciência da Computação, do Instituto de Informática (INF) da Universidade Federal de Goiás (UFG), como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Linha de pesquisa: Sistemas de Computação.

Orientadora: Profa. Dra. Sand Luz Corrêa

Co-Orientador: Prof. Dr. Kleber Vieira Cardoso

Goiânia
2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Cardoso, Pabllo Borges
Integração de uma Aplicação de Realidade Aumentada com Sistemas 5G seguindo o padrão 3GPP [manuscrito] / Pabllo Borges Cardoso. - 2024.
62, LXII f.: il.

Orientador: Profa. Dra. Sand Luz Corrêa; co-orientadora Dr. Kleber Vieira Cardoso.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2024.
Bibliografia.
Inclui siglas, abreviaturas, símbolos, gráfico, tabelas, lista de figuras, lista de tabelas.

1. 3GPP. 2. 5G. 3. CAPIF. 4. EDGE. 5. eXtended Reality. I. Luz Corrêa, Sand, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **43** da sessão de Defesa de Dissertação de **Pabllo Borges Cardoso**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos quatro dias do mês de dezembro de dois mil e vinte e quatro, a partir das nove horas, via webconferência, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Integração de uma Aplicação de Realidade Aumentada com Sistemas 5G seguindo o padrão 3GPP"**. Os trabalhos foram instalados pelo Coorientador, Professor Doutor Kleber Vieira Cardoso (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Leandro Alexandre Freitas (IFG), membro titular externo; Professor Doutor Antonio Carlos de Oliveira Junior (INF/UFG), membro titular interno. A realização da banca ocorreu por meio de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Kleber Vieira Cardoso, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos quatro dias do mês de dezembro de dois mil e vinte e quatro.
TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Antonio Carlos De Oliveira Junior**, **Professor do Magistério Superior**, em 04/12/2024, às 11:00, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Kleber Vieira Cardoso**, **Professor do Magistério Superior**, em 04/12/2024, às 11:01, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Leandro Alexandre Freitas**, **Usuário Externo**, em 04/12/2024, às 11:02, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Pabllo Borges Cardoso**, **Discente**, em 04/12/2024, às 11:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4992040** e o código CRC **2305F8AC**.

Referência: Processo nº 23070.056727/2024-07 SEI nº 4992040

Pabllo Borges Cardoso

Graduou-se em Análise de Sistemas pela Universidade Salgado de Oliveira (UNIVERSO) em 2002 e especializou-se em Tecnologia da Informação pela mesma instituição em 2004. Durante a graduação, trabalhou com o desenvolvimento de aplicações web com arquiteturas distribuídas, e, na especialização, atuou com processos e padrões de qualidade e maturidade de software, como CMM, CMMI, SPICE e MPS.br. No Mestrado pela UFG, atuou como pesquisador no LABORA-INF, apoiando a implementação de projetos envolvendo a detecção de objetos utilizando redes neurais, análise dos padrões 3GPP para aplicações EDGE, streaming de mídia e aplicações XR. É chefe do Departamento de Produção de Software da Defensoria Pública do Estado de Goiás (DPE-GO) desde 2017. Foi docente na Universidade Estadual de Goiás (UEG) de 2006 a 2017 e, até 2023, em instituições privadas. Em 2020 e 2021, recebeu menção honrosa por serviços prestados durante a pandemia pela Câmara Legislativa de Goiânia e, em 2022, foi homenageado pelos relevantes serviços prestados à sociedade na DPE-GO pela Assembleia Legislativa do Estado de Goiás (ALEGO).

Dedico este trabalho à minha família, que sempre esteve ao meu lado em cada etapa desta jornada. À minha amada esposa e companheira de vida, Claudia Karine, pelo amor incondicional, paciência e incentivo contínuo. À minha filha, Anne Karoline, cuja doçura e apoio renovam minha determinação. E ao meu filho, Pedro, que com alegria e energia me inspiram a buscar sempre o melhor. Sem o amor, o carinho, a compreensão e a força de cada um de vocês, nada disso seria possível. Esta conquista é, acima de tudo, fruto do nosso amor e união.

Agradecimentos

Gostaria de expressar minha profunda gratidão à Professora Sand Luz Corrêa e ao Professor Kleber Vieira Cardoso, que, com suas orientações e apoio, foram fundamentais para a realização deste trabalho. Agradeço pela paciência, dedicação e pelos valiosos conhecimentos transmitidos ao longo desta árdua jornada. Suas orientações não apenas enriqueceram meu desenvolvimento acadêmico, como também me inspiraram a seguir em frente, sempre buscando a excelência.

Agradeço também à coordenação e à secretaria do Programa de Pós-Graduação em Ciência da Computação (PPGCC), aos diretores e coordenadores do INF, aos técnicos administrativos da UFG e a todos os professores que me auxiliaram ao longo do caminho. Também agradeço aos inspiradores amigos e colegas da Defensoria Pública do Estado de Goiás (DPE-GO), pelo apoio, pelas trocas e pela compreensão.

Meu sincero agradecimento aos amigos do Laboratório de Redes de Computadores e Sistemas Distribuídos (LABORA-UFG), um grupo ímpar de pesquisa com pessoas muito especiais. Agradeço pela colaboração, pela amizade e pelo ambiente de pesquisa tão enriquecedor que vocês proporcionaram.

Aos membros da banca examinadora, agradeço pelo tempo dedicado e pelas valiosas contribuições para o aprimoramento desta pesquisa.

Agradeço ao Grande Arquiteto do Universo, Deus, que me proporcionou saúde e serenidade ao longo da vida. Em especial, agradeço à minha amada esposa e companheira de vida, Claudia Karine, à minha querida filha, Anne Karoline, e ao meu amado filho, Pedro, pela compreensão, pelo incentivo, pelo carinho e pela força espiritual que me sustentaram ao longo do caminho. Sem vocês e sem a proteção divina, nada disso seria possível."

Nós só podemos ver um pouco do futuro, mas o suficiente para perceber que há muito a fazer.

Alan Mathison Turing,
Matemático britânico e pioneiro da computação, Inglaterra.

Resumo

Borges Cardoso, Pabllo. **Integração de uma Aplicação de Realidade Aumentada com Sistemas 5G seguindo o padrão 3GPP**. Goiânia, 2024. 62p. Dissertação de Mestrado. Programa de Pós-Graduação em Ciência da Computação, Instituto de Informática (INF), Universidade Federal de Goiás (UFG).

Baseando-se nos padrões definidos pelo 3rd Generation Partnership Project (3GPP), este trabalho valida o modelo de 5G Media Streaming (5GMS), utilizando como estudo de caso o protótipo MR-Leo, uma aplicação de Realidade Mista (RM) projetada para explorar as potencialidades dessas tecnologias em ambientes de alta exigência computacional. O estudo inicia-se com uma revisão dos avanços promovidos pelas redes 5G, enfatizando sua capacidade de oferecer conectividade de baixa latência, alta largura de banda e suporte a dispositivos heterogêneos em larga escala. Complementarmente, são discutidos os arcabouços CAPIF e SEAL, concebidos para facilitar a interoperabilidade e o gerenciamento de APIs na arquitetura 5G, embora reconhecidos por sua complexidade técnica e limitada adoção prática. A computação de borda é então investigada como um componente estratégico, capaz de aproximar recursos computacionais dos usuários finais, atenuando latências e melhorando o desempenho de algoritmos intensivos essenciais para aplicações RM. A validação do estudo proposto foi realizada em três cenários distintos: um ambiente controlado local, uma rede 5G emulada e uma callbox 5G real. A avaliação experimental demonstrou a superioridade do protocolo, combinado com compressão de vídeo, alcançando métricas consistentes, atendendo aos indicadores-chave de desempenho (KPIs) definidos na literatura. A análise qualitativa comparativa evidenciou compatibilidades significativas, além de lacunas, como a ausência de um componente funcional equivalente ao 5GMS Application Function (AF). Neste sentido, este trabalho apresenta contribuições importantes ao demonstrar a viabilidade técnica da entrega de serviços RM em redes 5G por meio de computação de borda.

Palavras–chave

3GPP, 5G, CAPIF, EDGE, eXtended Reality

Abstract

Borges Cardoso, Pabllo. **Integration of an Augmented Reality Application with 5G Systems following 3GPP Standards**. Goiânia, 2024. 62p. MSc. Dissertation. Programa de Pós-Graduação em Ciência da Computação, Instituto de Informática (INF), Universidade Federal de Goiás (UFG).

Based on the standards defined by the 3rd Generation Partnership Project (3GPP), this work validates the 5G Media Streaming (5GMS) model, using the MR-Leo prototype as a case study. MR-Leo is a Mixed Reality (MR) application designed to explore the potential of these technologies in high-demand computational environments. The study begins with a review of advancements enabled by 5G networks, emphasizing their ability to provide low-latency connectivity, high bandwidth, and support for heterogeneous devices at scale. Additionally, the frameworks CAPIF and SEAL are discussed as tools to facilitate interoperability and API management in the 5G architecture, though recognized for their technical complexity and limited practical adoption. Edge computing is then investigated as a strategic component capable of bringing computational resources closer to end users, reducing latencies and enhancing the performance of intensive algorithms critical for MR applications. The validation of the proposed study was carried out in three distinct scenarios: a local controlled environment, an emulated 5G network, and a real 5G callbox. Experimental evaluation demonstrated the superiority of the protocol combined with video compression, achieving consistent metrics that meet the key performance indicators (KPIs) defined in the literature. The comparative qualitative analysis highlighted significant compatibilities as well as gaps, such as the absence of a functional component equivalent to the 5GMS Application Function (AF). In this regard, this work makes important contributions by demonstrating the technical feasibility of delivering MR services on 5G networks through edge computing.

Keywords

3GPP, 5G, CAPIF, EDGE, eXtended Reality

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

3GPP: *3rd Generation Partnership Project*
4G: *4th Generation mobile network system*
4K: *UHD 3840x2160 pixels de resolução*
5G: *5th Generation Mobile Network System*
5GC: *5G Core*
5G-XR: *5G eXtended Reality*
5GMS: *Streaming de Mídia 5G*
8K: *SHV 7680x4320 pixels*
AEF: *API Exposing Function*
AF: *Application Function*
AMF: *API Management Function*
AMMF: *Access and Mobility Management Function*
ANATEL: Agência Nacional de Telecomunicações
APF: *API Publishing Function*
API: *Application Programming Interfaces*
AR: *Augmented Reality*
AS: *Application Server*
CAPIF: *Common API Framework*
CCF: *CAPIF Core Function*
CG: *Cloud Gaming*
CN: *Core Network*
CU: *Central Unit*
DU: *Distributed Unit*
EDGE: *Edge Computing*
eMBB: *Enhanced Mobile Broadband*
Gbps: *Gigabits por segundo*
gNodeB: *gNode Base Station*
IoT: *Internet of Things*
LGPD: Lei Geral de Proteção de Dados
MAR: *Mobile Augmented Reality*

Mbps: *Megabits por segundo*
MJPEG: *Motion Joint Photographic Experts Group*
mMTC: *Massive Machine Type Communication*
MNO: *Mobile Network Operators*
MR-Leo: *Protótipo de aplicação com realidade mista por streaming*
MR: *Mixed Reality*
NEF: *Network Exposure Function*
NEFSim: *NEF Simulator*
NFV: *Network Function Virtualization*
NG-RAN: *Next Generation Radio Access Network*
NS: *Network Slicing*
PCF: *Policy Control Function*
PLMN: *Public Land Mobile Network*
QoS: *Quality of Service*
RAN: *Radio Access Network*
RRU: *Remote Radio Unit*
SA2: *System Architecture and Services*
SA4: *Multimedia Codecs, Systems and Services*
SBA: *Service-Based Architecture*
SDN: *Software Defined Network*
SEAL: *Service Enabler Layer Architecture*
SHV: *Super Hi-Vision*
SMF: *Session Management Function*
SVE: *Shared Virtual Environment*
UE: *User Equipment*
UERANSIM: *User Equipment and Radio Access Network Simulator*
UDP: *User Datagram Protocol*
UHD: *Ultra HD (Digh Definition)*
UPF: *User Plane Function*
URLLC: *Ultra-reliable Low Latency Communication*
VAC: *Vertical Application Client*
VAE: *Vertical Application Enablers*
VR: *Virtual Reality*
XR: *Extended Reality*

CAPÍTULO 1

Introdução

Com a consolidação da especificação das redes móveis de 5ª geração (*5th Generation Mobile Network System* - 5G), padronizada principalmente pelo *3rd Generation Partnership Project* (3GPP) nos documentos [3GPP 2019, 3GPP 2020, 3GPP 2022-a], o próximo passo natural é a implantação do 5G. Nesse sentido, esta tecnologia está sendo implantada em todo mundo para proporcionar um salto significativo em relação à arquitetura das redes móveis de 4ª geração (*4th Generation Mobile Network System* - 4G), oferecendo muito mais velocidades de conexão, maior capacidade, latência ultra baixa e suporte para uma ampla gama de aplicações e dispositivos. No Brasil, a Agência Nacional de Telecomunicações (ANATEL) realizou, em novembro de 2021, um leilão que definiu quais as operadoras de redes móveis (*Mobile Network Operators* - MNOs) operarão o 5G no país. A estimativa inicial é que todo território brasileiro esteja coberto até 2029. De fato, uma característica distintiva do 5G é a sua possibilidade de modificar a relação que o usuário tem atualmente com as aplicações móveis, propiciando maior qualidade de experiência para esses usuários. Neste sentido, o 5G define três casos de uso principais: *Enhanced Mobile Broadband* (eMBB), compreendendo serviços que requerem alta largura de banda, como aplicações de realidade aumentada (*Augmented Reality* - AR), realidade virtual (*Virtual Reality* - VR) e *streaming* de vídeo em 360 graus; *Ultra-reliable Low Latency Communication* (URLLC), englobando aplicações de missão crítica que requerem alta disponibilidade e baixa latência, como carros autônomos e cirurgia remota; e *Massive Machine Type Communication* (mMTC), compreendendo serviços que envolvam um grande número de dispositivos conectados simultaneamente, como aplicações para Internet das Coisas (IoT) [Gupta e Jha 2015].

Para atender todos esses casos de usos com objetivos diversos e de forma flexível, o 5G faz amplo uso de software em diferentes níveis e componentes de sua arquitetura [Both et al. 2020]. Um exemplo é a rede de núcleo (*core network*), ou núcleo 5G, que expõe à terceiros, de forma segura, um conjunto de interfaces (*Application Programming Interfaces* - APIs) bem definidas e padronizadas. Terceiros, devidamente autorizados, como indústrias, desenvolvedores de plataformas e projetista, podem usar essas APIs para criar aplicações cientes da rede, à quais estabelecem uma comunicação

bi-direcional com o núcleo para recuperar estatísticas da rede, bem como acionar políticas e comandos específicos. Essa capacidade de exposição proporcionada pelo núcleo 5G é materializada através de uma arquitetura baseada em serviços (*Service-Based Architecture* - SBA), onde as funções de rede do plano de controle se comunicam por meio de chamadas para APIs.

Neste contexto, uma função do plano de controle do núcleo que se torna importante é a *Network Repository Function* (NRF). Essa função permite que outras funções de rede registrem seus serviços, os quais podem, posteriormente, serem descobertos e consumidos por outras funções. Outra função relevante é a *Network Exposure Function* (NEF), a qual fornece adaptadores para conectar as interfaces *southbound* da SBA à uma camada de exposição formada por interfaces *northbound* oferecidas a terceiros. Dessa forma, a NEF facilita a divulgação segura de recursos de rede para terceiros, como fatiamento de rede (*network slicing*), computação de borda (*edge computing*) e aprendizado de máquina, permitindo, por exemplo, a monetização de ativos de rede e nova oportunidades de negócios para as operadoras. Portanto, as funcionalidades fornecidas pela NRF e NEF a terceiros criam um novo ecossistema onde código de terceiros, devidamente autorizados, podem combinar as capacidades expostas pelo núcleo com os requisitos de aplicações, promovendo maior inovação e colaboração.

Para padronizar como serviços de terceiros têm acesso às capacidades expostas pela rede 5G, o 3GPP definiu dois métodos. O primeiro, denominado *Common API Framework* (CAPIF), foi definido durante a elaboração do Release 15 [3GPP 2019], visando definir uma abordagem unificada entre a API *northbound* do núcleo 5G (5G *core* - 5GC) e as aplicações de terceiros. Neste contexto, o objetivo do CAPIF é padronizar as funcionalidades comumente suportadas pela API *northbound* do 5GC, como autenticação, descoberta de serviços e políticas de cobrança, para facilitar o desenvolvimento de aplicações de terceiros.

No entanto, à medida que a demanda por padrões para desenvolver e implantar aplicações nas redes 5G para diferentes setores produtivos aumenta, ficou evidente que muitos serviços são comuns a várias aplicações, como, por exemplo, o serviço de gerenciamento de localização. Para evitar a duplicação de esforços e oferecer tais serviços como uma camada na arquitetura da rede, o 3GPP definiu, durante a especificação do Release 16 [3GPP 2020], o segundo método, denominado *Service Enabler Architecture Layer* (SEAL). O SEAL permite que esses serviços comuns sejam consumidos pelas aplicações de terceiros por meio de APIs compatíveis com o CAPIF. Essa abordagem traz benefícios tanto para a indústria vertical, que passa a ter acesso a serviços auxiliares padronizados, evitando a necessidade de desenvolvê-los internamente, quanto para os MNOs, que podem evitar a duplicação de esforços e oferecer uma implantação mais eficiente para as aplicações de terceiros.

Na prática, no entanto, o uso do CAPIF e do SEAL tem se mostrado bastante complexo, tendo sido validado em poucos trabalhos [Sanchez et al. 2022, Tsolkas e Koumaras 2022]. Essa dificuldade foi evidenciada durante a elaboração do Release 17 [3GPP 2022-a], quando o 3GPP apresentou uma arquitetura simplificada para disponibilizar aplicações de realidade estendida (*Extended Reality* - XR) em redes 5G.

De fato, aplicações XR têm ganhado bastante destaque nos últimos anos, oferecendo uma variedade de possibilidades e transformando a maneira como interagimos com o mundo digital e físico. XR engloba tecnologias como VR e AR, proporcionando experiências imersivas e interativas que vão além da realidade convencional [Gapeyenko et al. 2023]. As aplicações XR têm encontrado espaço inovador em diversos segmentos [Taleb et al. 2023]. Na indústria automotiva, por exemplo, XR é utilizada para projetar e simular veículos, permitindo que engenheiros e projetistas visualizem modelos virtuais em escala real e realizem testes virtuais de segurança. Na indústria da construção, XR possibilita a visualização de projetos em 3D e a detecção de erros antes mesmo do início da construção física. XR também é aplicada em treinamentos de funcionários, oferecendo simulações realistas e interativas para aprimorar habilidades e segurança no trabalho. Na área da saúde, XR permite simulações médicas, treinamentos de cirurgias e terapias de reabilitação imersivas. Os profissionais de saúde podem praticar procedimentos complexos em ambientes virtuais antes de realizá-los em pacientes reais, aumentando a precisão e a segurança dos procedimentos médicos.

A padronização do suporte a aplicações XR sobre um sistema 5G começou no 3GPP em 2016, com o grupo de trabalho *Service and System Aspect* especificando requisitos de serviço 5G para aplicações XR de alta taxa e baixa latência [3GPP 2021-a]. O trabalho continuou em 2018, documentando características relevantes do tráfego e fornecendo um levantamento de aplicações XR [3GPP 2021-b]. Paralelamente, novos identificadores de qualidade de serviço 5G foram padronizados para oferecer suporte a serviços interativos, incluindo XR [3GPP 2021-c]. Recentemente, o esforço continuou com [3GPP 2022-b], onde o 3GPP definiu uma arquitetura simplificada para disponibilizar aplicações XR sobre um sistema 5G, utilizando o modelo de *streaming* de mídia para o 5G. Nesse modelo, um provedor de aplicações de XR faz uso das funções do 5GC por meio de três módulos principais: uma *Application Function* (AF), dedicada para serviços XR; um servidor de aplicação, dedicado para serviços XR; e um módulo cliente, ciente da rede e dos serviços XR disponíveis, o qual executa no dispositivo do usuário (*User Equipment* - UE). Apesar de relativamente simples, até onde sabemos, esse modelo ainda não foi validado na prática.

Neste contexto, o objetivo geral deste trabalho é investigar como uma aplicação XR pode ser integrada a uma rede 5G, utilizando computação de borda. Particularmente, estendemos um protótipo XR, que faz uso de computação de borda, existente na literatura

para executar em uma rede 5G. Avaliamos, então, esta integração de duas forma: qualitativamente, comparando-a com o modelo de *streaming* de mídia para o 5G do 3GPP; e quantitativamente por meio de um ambiente emulado e um *setup* 5G real. Comparamos também o desempenho do protótipo XR em três situações: (1) em um *setup* WiFi; (2) usando o ambiente emulado; e (3) usando o *setup* real. Para atingir esse objetivo, no ambiente emulado, utilizamos uma implementação de código aberto de uma 5GC, denominada free5GC [free5GC 2024], projetada para ser compatível com o padrão 5G definido pela 3GPP e amplamente usado por pesquisadores, desenvolvedores e engenheiros para prototipar redes 5G, sem a necessidade de investir em infraestruturas caras de telecomunicação. Utilizaremos também o (*User Equipment and Radio Access Network Simulator* - UERANSIM) [UERANSIM 2024], uma ferramenta de código aberto usada para emular o comportamento de UEs e redes de acesso via rádio (*Radio Access Network* - RAN) em ambientes de teste 5G. Esse emulador também é amplamente utilizado em conjunto com o 5GC, emulado pelo free5GC, para criar um ambiente completo de teste para redes 5G. Para o *setup* real usamos uma estação rádio base, de uso experimental, produzida pela Amarisoft que oferece comunicação 4G e 5G. Finalmente, para a aplicação XR, utilizaremos o *Mixed Reality Linköping Edge Offloading* (MR-Leo) [Toczé e et al. 2019], um protótipo originado de uma dissertação de mestrado que explora a utilização da computação de borda para habilitar serviços de AR em dispositivos móveis.

Esta dissertação está organizada conforme a estrutura descrita a seguir.

- No Capítulo 2, apresentamos a fundamentação teórica no qual este trabalho se baseia, incluindo uma visão geral da arquitetura 5G, computação de borda e realidade estendida e discutimos o relacionamento entre essas tecnologias. Apresentamos também os principais trabalhos relacionados e as diferenças em relação a este trabalho.
- No Capítulo 3, apresentamos o modelo de *streaming* de mídia para o 5G proposto pelo 3GPP. Introduzimos também o protótipo MR-Leo utilizado neste trabalho. Fazemos uma análise qualitativa do quanto a arquitetura do protótipo pode ser mapeada para a arquitetura do modelo proposto pelo 3GPP. Por fim, apresentamos também uma análise de desempenho do protótipo em diferentes cenários e *setups*.
- O Capítulo 4 apresenta as conclusões deste trabalhos e direções para trabalhos futuros.

CAPÍTULO 2

Fundamentação Teórica e Trabalhos Relacionados

Este capítulo fornece uma visão geral dos fundamentos e das principais tecnologias abordadas nesta dissertação. Neste sentido, inicialmente, apresentamos a arquitetura das redes 5G, conforme definido pelo 3GPP [3GPP 2017] (Seção 2.1). Em seguida, discutimos o paradigma de computação de borda e seu relacionamento com as redes móveis de 5ª geração (Seção 2.2). Os fundamentos de XR são apresentados na Seção 2.3, enquanto na Seção 2.4 discutimos a relação entre 5G, computação de borda e XR. Examinamos os trabalhos mais relevantes que se relacionam diretamente com este estudo na Seção 2.5, enquanto a Seção 2.6 traz as considerações finais deste capítulo.

2.1 Visão geral da arquitetura 5G

A tecnologia 5G oferece "velocidade" (ou taxa de dados) de 10 a 100 vezes maior que a alcançada nas redes 4G, sendo capaz de conectar um número de dispositivos por Km² até 100 vezes maior que a geração anterior [Gupta e Jha 2015]. Além disso, as redes 5G podem apresentar uma latência fim-a-fim de apenas 5 ms (milessegundos), a qual é apenas uma fração da latência típica em redes sem fio atuais [Khalili et al. 2018].

Além de maior taxa de dados, maior conectividade e latência de comunicação reduzida, o 5G introduz várias novidades em relação à geração anterior. Por exemplo, na tecnologia 5G, o núcleo da rede é desagregado em diversas funções, seguindo uma arquitetura baseada em serviços, onde as funções de rede são implementadas como serviços independentes que se comunicam através de APIs padronizadas. O núcleo 5G utiliza também tecnologias e padrões de virtualização, como virturalização de funções de rede (*Network Functions Virtualization* - NFV), redes definidas por software (*Software-Defined Networking* - SDN) e fatiamento de rede (*Network Slicing*) [Foukas et al. 2017]. Essas tecnologias de virtualização são fundamentais para que redes 5G possam atender de maneira adequada serviços com requisitos muito distintos, como serviços que requerem alta largura de banda (eMBB), aplicações de missão crítica que requerem alta

disponibilidade e baixa latência (URLLC) e serviços que envolvam um grande número de dispositivos conectados simultaneamente (mMTC).

A arquitetura das redes 5G é especificada no Release 15 do 3GPP [3GPP 2017]. Considerando especificamente a transição entre 4G e 5G, o documento apresenta duas arquiteturas: Não-Autônoma (*Non-Stand Alone* - NSA) e Autônoma (*Stand-Alone* - SA). A arquitetura NSA foi proposta em maio de 2018 e, seis meses depois, o 3GPP apresentou as especificações para a arquitetura SA. A primeira (NSA) é uma forma rápida de prover alta vazão de dados através do aproveitamento dos ativos de rede existentes, isto é, sem a necessidade de implantar um novo sistema completo de ponta-a-ponta para a rede 5G. Dessa forma, a NSA consiste em uma arquitetura de transição da 4G para 5G, onde somente a tecnologia de radio precisa ser atualizada. Por outro lado, a arquitetura SA define uma solução independente do sistema 4G, sendo considerada, portanto, a arquitietura 5G definitiva. Neste trabalho, consideraremos apenas a arquitetura SA, a qual é melhor descrita nas subseções seguintes.

2.1.1 Arquitetura 5G Autônoma (SA)

Como ilustrado na Figura 2.1, a arquitetura 5G SA é composta basicamente por duas partes essenciais: o núcleo 5G (5GC) e a rede de acesso via rádio (RAN).

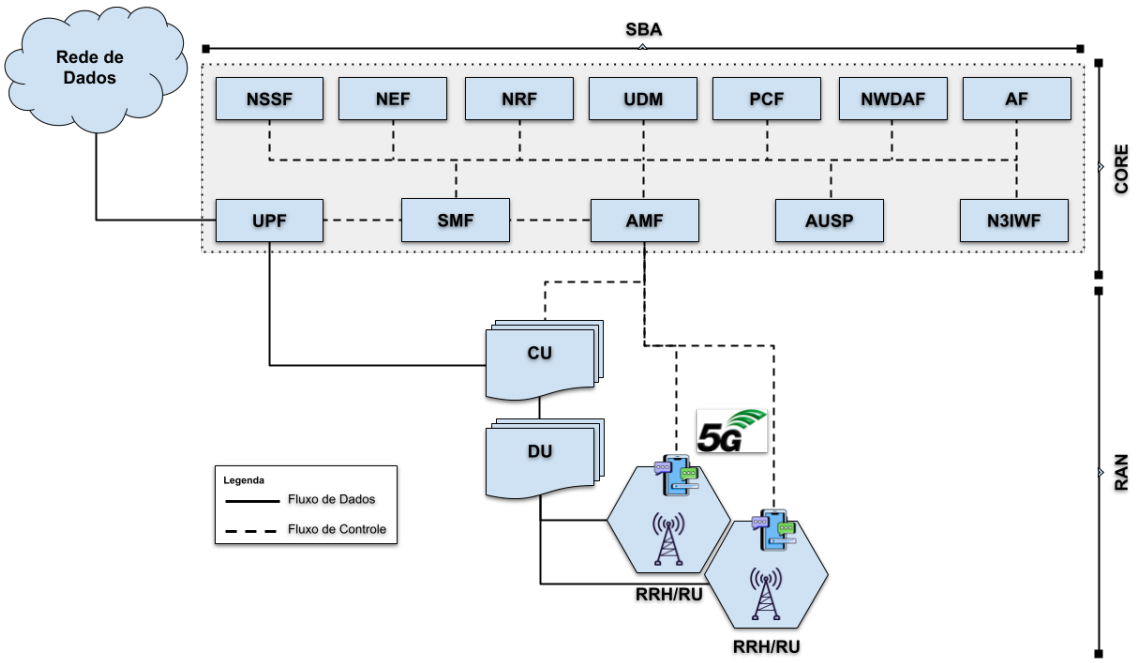

Figura 2.1: *Arquitetura Baseada em Serviço (SBA) [3GPP 2017]*

O *5GC* [Sanchez et al. 2022] é o núcleo da rede, sendo a parte central da arquitetura 5G. Ele é responsável por gerenciar o tráfego de dados e de controle entre

dispositivos de usuário (UE) e as redes externas, além de suportar grande parte das funcionalidades avançadas que tornam o 5G mais flexível.

De fato, o 5GC é a parte da arquitetura 5G mais softwerizada, sendo projetado como um barramento de serviços integrados (SBA) que divide as funções de rede em entidades lógicas independentes, com baixo acoplamento. Essas entidades são denominadas *Network Functions* (NFs) e se comunicam entre si por meio de interfaces padronizadas e pontos de referência, conforme ilustrado na Figura 2.2. A SBA é construída com base em tecnologias de virtualização e microserviços, tornando o núcleo 5G altamente escalável e flexível. Além disso, no 5GC, o plano de controle e o plano de usuário (dados) são implementados e gerenciados de forma separada, melhorando a eficiência e o gerenciamento da rede. As NFs especificadas no release 15 do 3GPP são [3GPP 2017, Gupta e Jha 2015]:

- *Access and Mobility Management Function* (AMF): gerencia a autenticação, mobilidade e registro dos UEs;
- *Session Management Function* (SMF): controla a criação e o gerenciamento de sessões de dados;
- *User Plane Function* (UPF): encaminha pacotes de dados entre a rede e os UEs;
- *Unified Data Management* (UDM): armazena informações de assinantes e configurações;
- *Policy Control Function* (PCF): define e aplica políticas de qualidade de serviço (QoS) e gerenciamento de tráfego;
- *Authentication Server Function* (AUSF): realiza autenticação dos UEs conectados à rede;
- *Network Slice Selection Function* (NSSF): seleciona e gerencia fatias de rede para diferentes aplicações; e
- *Network Exposure Function* (NEF): expõe APIs para serviços externos e aplicações de terceiros;

A Figura 2.3 apresenta uma compilação das terminologias utilizadas na arquitetura SBA, as NFs, as interfaces e seus respectivos pontos de referência.

Como mostrado na Figura 2.1, a outra parte essencial da arquitetura 5G é a RAN, responsável por estabelecer a comunicação sem fio entre o UE e o núcleo da rede, incluindo a transmissão, recepção e gerenciamento do tráfego de dados e controle no lado do rádio [Sanchez et al. 2022]. No 5G, o espectro de frequências disponível para a RAN foi ampliado não apenas em bandas de sub-6GHz, mas também na faixa de ondas milimétricas, ou seja, de dezenas de GHz.

Assim como o núcleo, a estação rádio base 5G, denominada gNodeB, também pode ser virtualizada, levando ao conceito de *virtualized* RAN ou vRAN [Garcia-Saavedra e Costa-Pérez 2021]. A vRAN tem recebido destaque nas redes 5G, pois permite criar,

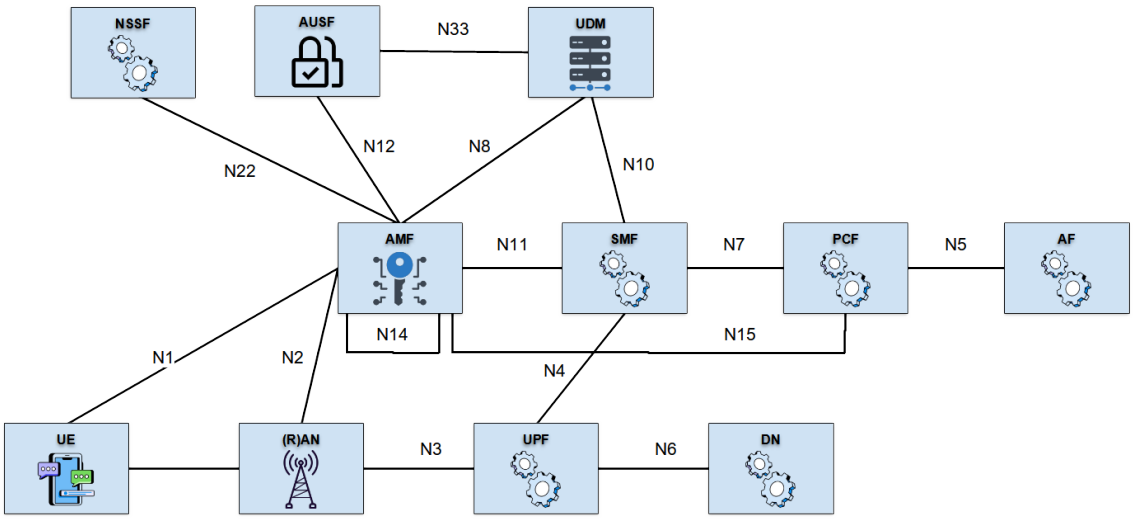

Figura 2.2: *Pontos de referência da arquitetura [3GPP 2017]*

gerenciar e configurar RANs dinamicamente, atendendo requisitos específicos de cada serviço. Particularmente, a gNodeB pode ser desagregada em até três entidades, a saber: *Radio Unit* (RU), *Distributed Unit* (DU) e *Centralized Unit* (CU). As funções executadas em cada uma dessas entidades são determinadas pelo tipo de desagregação (*split*) escolhido, sendo definidos pelo 3GPP oito diferentes tipos [Larsen, Checko e Christiansen 2019].

2.1.2 Camadas habilitadoras de serviço para verticais

Os MNOs estão fazendo grandes investimentos para implantar as redes 5G. No entanto, obter retorno de investimento apenas por meio de usuários finais está se tornando cada vez mais difícil [Shah et al. 2020]. Assim, o 5G foi projetado para ter recursos avançados como fatiamento de rede e computação de borda, considerando requisitos de segmentos ou indústrias verticais, incluindo saúde, setor automotivo, fábricas inteligentes, comunicações de missão crítica, etc. Para permitir uma implantação mais rápida de serviços verticais é necessário padronizar como esses serviços podem requisitar/acessar funcionalidades de uma rede 5G.

Um dos primeiros esforços do 3GPP neste sentido foi feito com a especificação do arcabouço *Common API Framework* (CAPIF) [3GPP 2020]. O CAPIF é um arcabouço padrão criado pelo 3GPP para fornecer um ponto comum de acesso às APIs de rede (*Northbound APIs*), utilizadas para interagir com as funções do núcleo 5G. De fato, algumas funções de rede do núcleo, como autenticação, descoberta de serviços e políticas de cobrança/bilhetagem, são comumente consumidas por aplicações verticais.

A Figura 2.4 apresenta uma visão geral do arcabouço, o qual é formado por três componentes principais: *API Invoker*, *CAPIF Core Function* e *API provider* [Charismiadis et al. 2023]. O componente *API Invoker* é normalmente fornecido por uma aplicação

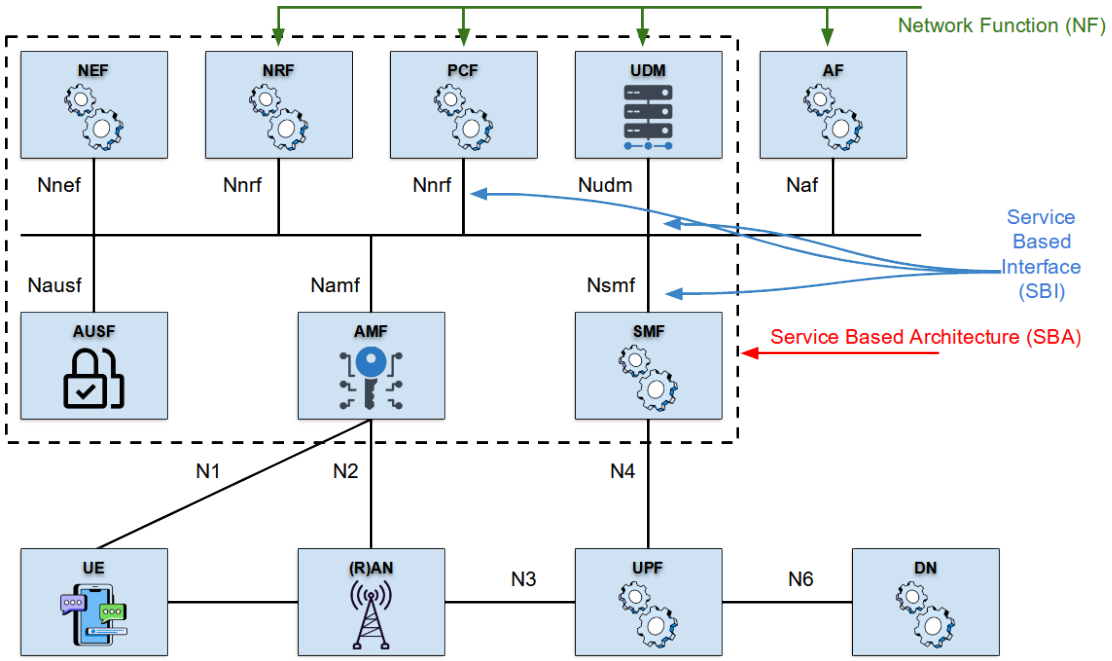

Figura 2.3: *Terminologias utilizadas da SBA [3GPP 2017]*

de terceiros que precisa consumir serviços da rede. O componente *CAPIF Core Function* (CCF) é a entidade principal do arcabouço, sendo responsável por: autenticar um *API Invoker* com base na identidade e/ou outras informações; autorizar um *API Invoker* antes de acessar APIs de serviço; fazer o *onboard*/*offboard* de *API Invokers*; monitorar invocações de APIs de serviço; e armazenar configurações de política relacionadas ao CAPIF e às APIs de serviço. O componente *API provider* é uma entidade que fornece funções de exposição, publicação e gerenciamento de APIs. Tendo como base esses três componentes, um conjunto de pontos de referência também são especificados, como mostrado na Figura 2.4. As interfaces CAPIF-1 a CAPIF-5 podem ser usadas por *API invokers*/*API providers* dentro do mesmo domínio *Public Land Mobile Network* (PLMN) onde o CCF está localizado, enquanto as interfaces CAPIF-1e e CAPIF-2e são usadas por *API invokers*/*API providers* que estão fora do domínio PLMN do CCF.

Enquanto a demanda para desenvolver padrões de aplicações verticais para diferentes tipos de indústrias estava aumentando continuamente, tornou-se óbvio que muitos serviços auxiliares, como gerenciamento de localização, são necessários em vários verticais. Como resultado, capturar esses serviços auxiliares comumente usados e oferecê-los como uma camada de serviço comum para terceiros pode ajudar tanto as indústrias verticais (permitindo que eles se concentrem apenas nas funcionalidades de suas aplicações) quanto aos MNOs, poupando-os de enormes esforços e tempo para desenvolver os serviços correspondentes para cada vertical. O conceito acima tornou-se realidade com a definição do *Service Enabler Layer Architecture* (SEAL) no Release 16 [3GPP 2020]. O SEAL permite que esses serviços comuns sejam consumidos pelos

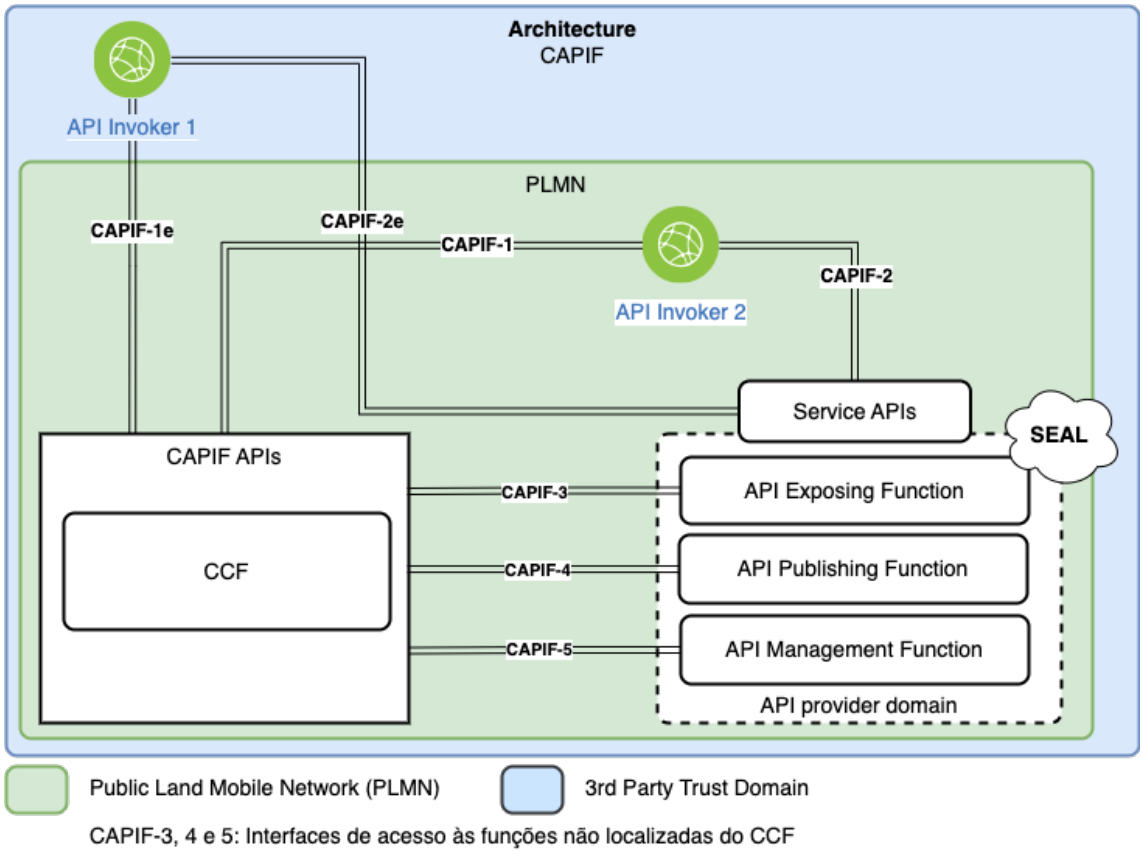

Figura 2.4: *Arquitetura simplificada do CAPIF [3GPP 2020]*

verticais por meio de APIs *northbound* e o CAPIF. A arquitetura suporta dois modelos funcionais: *On-network*, usando a interface SEAL-Uu, quando o UE se conecta à rede 5G para consumir o serviço, e *Off-network*, quando os UEs se conectam diretamente entre si.

A Figura 2.5 ilustra a arquitetura SEAL, considerando o modelo *On-network*. O componente *Vertical Application Layer Client* (VAL *client*) provê as funcionalidades do lado cliente da aplicação vertical, enquanto o componente *Vertical Application Layer Server* (VAL *server*) provê as funcionalidades do lado servidor da aplicação. Se usado com o CAPIF, o VAL *server* atua como uma função de exposição de APIs para o servidor de aplicação do vertical (*Vertical Application Server*) ou outro *Vertical Application Server* (VAE). Ele também pode atuar como um *API Invoker* para consumir as APIs de serviço de outro VAL *server*. Finalmente, os componentes *SEAL Client* e *SEAL Server* proveem as funcionalidades do lado cliente e servidor, respectivamente, de um serviço SEAL (por exemplo, o lado cliente/servidor de um serviço de localização) [Fragkos et al. 2021].

Embora o uso de arcabouços como CAPIF e SEAL trazer benefícios significativos para a interoperabilidade e integração de serviços na arquitetura 5G, na prática, existem algumas desafios que podem surgir na adoção e implementação desses arcabouços [Fragkos et al. 2021]. Em geral, a curva de aprendizado é grande uma vez que operadores, desenvolvedores e integradores de sistemas precisam entender as especificações

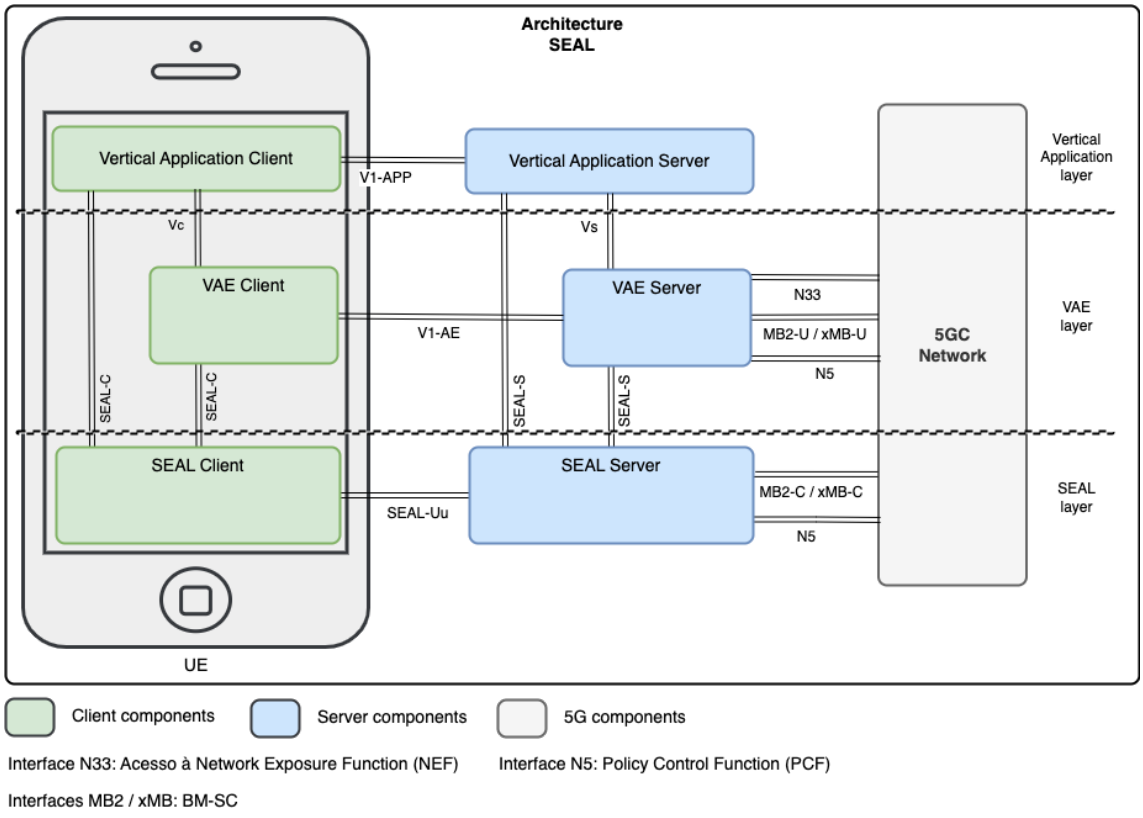

Figura 2.5: *Arquitetura simplificada do SEAL [3GPP 2020]*

3GPP e a funcionalidade dos arcabouços, levando tempo e esforço significativo. A manutenção dos arcabouços também é desafiadora, uma vez que exige monitoramento contínuo para garantir conformidade com os padrões e atender a novos requisitos de casos de uso.

2.2 Computação de borda

O aumento de dispositivos sem fio conectados à Internet, tem motivado o surgimento de novos serviços e aplicações que exigem recursos abundantes de processamento e/ou armazenamento. Como a grande maioria dos dispositivos sem fio possuem restrições de recursos e de bateria, uma solução natural é distribuir as funcionalidades dessas novas aplicações entre o dispositivo sem fio e servidores localizados na nuvem. Esses servidores, no entanto, geralmente encontram-se topologicamente distantes dos dispositivos, fazendo com que a requisição percorra caminhos diferentes até chegar no servidor. Como consequência, a distância percorrida ou a complexidade do caminho em função da quantidade de saltos, pode ocasionar ineficiência da aplicação ou até uma experiência ruim para o usuário. Neste caso, para aplicações que possuem restrições severas de latência [Shi et al. 2016], a solução mais adequada é servi-las usando recursos de processamento e/ou armazenamento localizados próximos aos usuários finais. Esse paradigma de computação é denominado Computação de Borda (*Edge Computing*) [ETSI 2016].

A computação de borda transforma significativamente a infraestrutura de telecomunicações, trazendo processamento e armazenamento para mais próximo dos usuários finais, o que resulta em redução drástica de latência e melhoria na eficiência da rede. Essa proximidade é importante para aplicações que demandam resposta rápida, como realidade aumentada, sistemas autônomos e jogos online, além de otimizar o desempenho de dispositivos IoT. A computação de borda não só aprimora a experiência do usuário por meio de serviços mais ágeis e personalizados, mas também possibilita o surgimento de novos modelos de negócios, ao mesmo tempo que contribui para a redução do consumo de energia e custos operacionais. Além disso, processar dados na origem minimiza riscos de segurança, permitindo a implementação de protocolos de proteção mais efetivos e adaptados às necessidades.

Os princípios da computação de borda remontam ao final da década de 1990, com a introdução das redes de distribuição de conteúdo (*Content Delivery Network* - CDN). Essas redes introduzem nós de armazenamento próximos aos usuários finais, geralmente visando melhorar o desempenho de aplicações Web, especialmente aquelas voltadas para transmissão de vídeos. Como nas CDNs, na computação de borda, recursos de processamento e/ou armazenamento são inseridos próximos aos usuários finais para servir aplicações que rodam nos dispositivos sem fio, diminuindo a latência de comunicação. Contudo, diferentemente dos nós CDN, os recursos de borda não estão restritos ao armazenamento de conteúdo, podendo oferecer funcionalidades de processamento e de gerenciamento de serviços similares às encontradas em sistemas de computação em nuvem. Dessa forma, a computação de borda foi idealizado não apenas para permitir a execução de serviços tradicionais, como cache de vídeo, mas também novos serviços como os demandados pela Indústria 4.0, cirurgia remota, robôs colaborativos, carros autônomos, XR e jogos online.

A princípio, recursos de borda podem ser alocados de forma co-localizada com estações base, ou com algum equipamento da RAN, ou ainda no próprio núcleo da rede [Contreras et al. 2020]. No entanto, à medida que se aproximam dos dispositivos sem fio, os recursos de borda tendem a se tornar cada vez mais escassos e caros.

Outra vantagem da computação de borda é que os recursos de processamento e/ou armazenamento geralmente estão mais integrados à infraestrutura de rede, diferentemente de outros paradigmas que se baseiam em computação próxima ao usuário final, como a Computação em Névoa (*Fog Computing*) e *Cloudlets*. Como consequência, a computação de borda é mais consciente dos recursos da rede em comparação com outros paradigmas [Cruz, Achir e Viana 2022]. Isso facilita, por exemplo, a infraestrutura de gerenciamento a obter informações em tempo real e provê-las às aplicações.

A arquitetura de referência para a computação de borda, ilustrada na Figura 2.6, foi especificada pela *European Telecommunications Standards Institute* (ETSI) no

GS MEC 003 V1.1.1 de 2016 [ETSI 2016]. Essa arquitetura é referenciada como *Multi-access Edge Computing* (MEC), sendo dividida em dois níveis: *MEC System* e *MEC Host*. O nível *MEC System* consiste em um conjunto de entidades que gerenciam o nível *MEC Host*. Essas entidades são descritas a seguir:

- *Device Application*: qualquer aplicativo que execute no dispositivo do usuário e que seja capaz de interagir com o sistema MEC. Essa entidade interage com o *Lifecycle Management (LCM) proxy* para solicitar a instanciação de uma aplicação MEC. No contexto da arquitetura, uma aplicação MEC é aquela instanciada em uma infraestrutura de computação de borda para atender requisições de dispositivos de usuários.
- *LCM Proxy*: entidade que recebe as solicitações das *Device Applications* para acionar a instanciação, encerramento e realocação de aplicações MEC. Esta entidade também expõe, para as *Device Applications*, informações sobre o estado da aplicação MEC.
- *Operations support system* (OSS): essa entidade recebe, do *LCM proxy*, os pedidos das *Device Applications*. Ela, então, decide se deve conceder (ou não) a solicitação, com base nas políticas do MNO.
- *Multi-access Edge Orchestrator* (MEO): é a entidade que mantém uma visão global do sistema MEC. Ela recebe a solicitação de criação da aplicação MEC e escolhe em qual (ou quais) *MEC host*(s) alocar a aplicação, sendo responsável por iniciar a instanciação, encerramento e realocação de aplicações MEC.

O nível *MEC Host* contém as entidades que executam ou gerenciam a infraestrutura de borda. Essas entidades são descritas a seguir.

- *MEC host*: entidade que provê recursos de computação, armazenamento e rede para aplicações MEC.
- *MEC platform*: esta entidade executa em cada *MEC host*, oferecendo um serviço de registro de forma que aplicações possam anunciar, descobrir e consumir serviços MEC. É acionada pela entidade *MEC platform manager* para instanciar uma aplicação ou serviço MEC e configurar o plano de dados da aplicação ou serviço instanciado.
- *MEC platform manager*: entidade única que executa duas tarefas: a gerência do ciclo de vida das aplicações MEC que executam nos *MEC hosts* e a gerência das plataformas MEC. É acionada pelo MEO para acionar a entidade *MEC platform* para instanciar (ou encerrar) uma aplicação MEC.
- *Virtualization infrastructure manager* (VIM): entidade única que gerencia a virtualização dos recursos dos *MEC hosts*, aplicando as regras definidas pelo *MEC platform manager*.

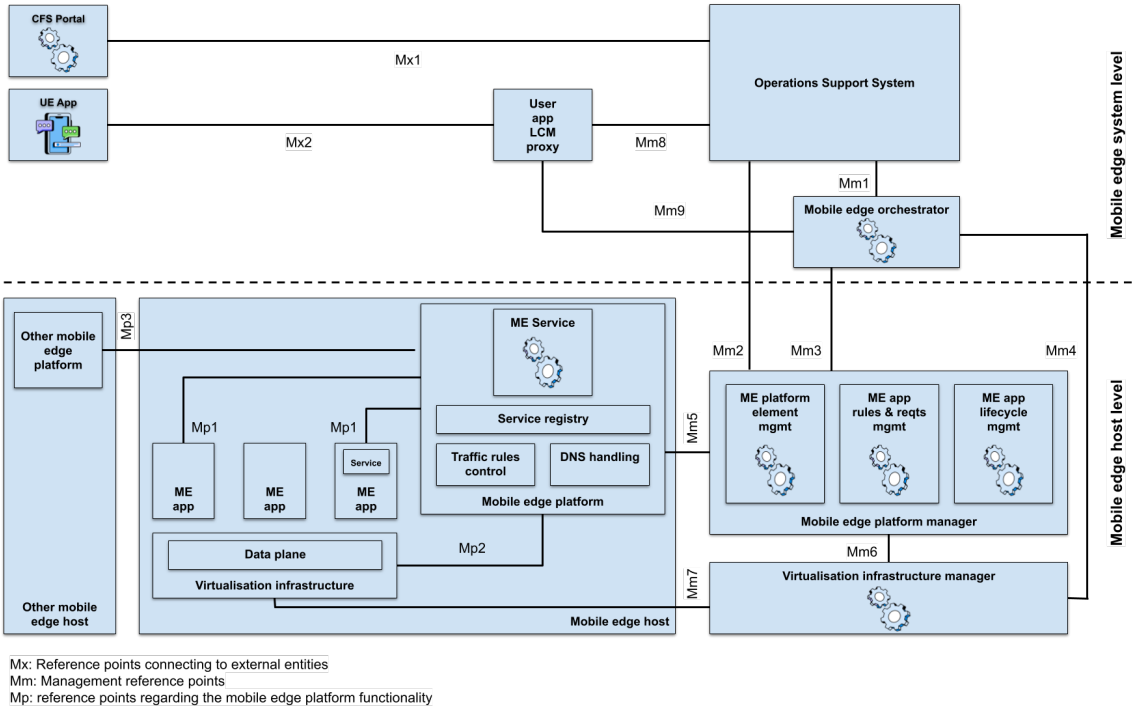

Figura 2.6: *Arquitetura de referência MEC*

É importante ressaltar que existem sobreposições de funções entre a arquitetura MEC e os arcabouços CAPIF e SEAL. Isso ocorre porque todas essas arquiteturas fornecem APIs para que aplicações externas interajam com a infraestrutura da rede 5G. No entanto, como a arquitetura MEC foi definida pela ETSI e os arcabouços CAPIF e SEAL foram especificados pelo 3GPP, não existe ainda uma visão clara de como alguns componentes da arquitetura MEC podem mapear/substituir componentes dos outros arcabouços.

2.3 Realidade estendida

A Realidade Estendida - *eXtended Reality* (XR) é uma forma de interação do homem com a máquina que combina ambientes reais e virtuais através do uso de recursos computacionais [Huang et al. 2023]. As primeiras interações baseadas em XR eram bem limitadas. No entanto, com a evolução dos dispositivos móveis, dispostivos IoT e vestíveis, é possível, atualmente, atingir um nível de experiência bem mais realista. Contudo, isso exige uma aumento significativo na necessidade de processamento e armazenamento, bem como baixa latência.

O 3GPP, no TR 26.928 Release 18 [3GPP 2023], faz referência ao termo XR como um concentrador de todas as tecnologias imersivas, seja Realidade Virtual - *Virtual Reality* (VR), Realidade Aumentada - *Augmented Reality* (AR), ou Realidade Mista - *Mixed Reality* (MR), cada uma com características específicas para a experiência do

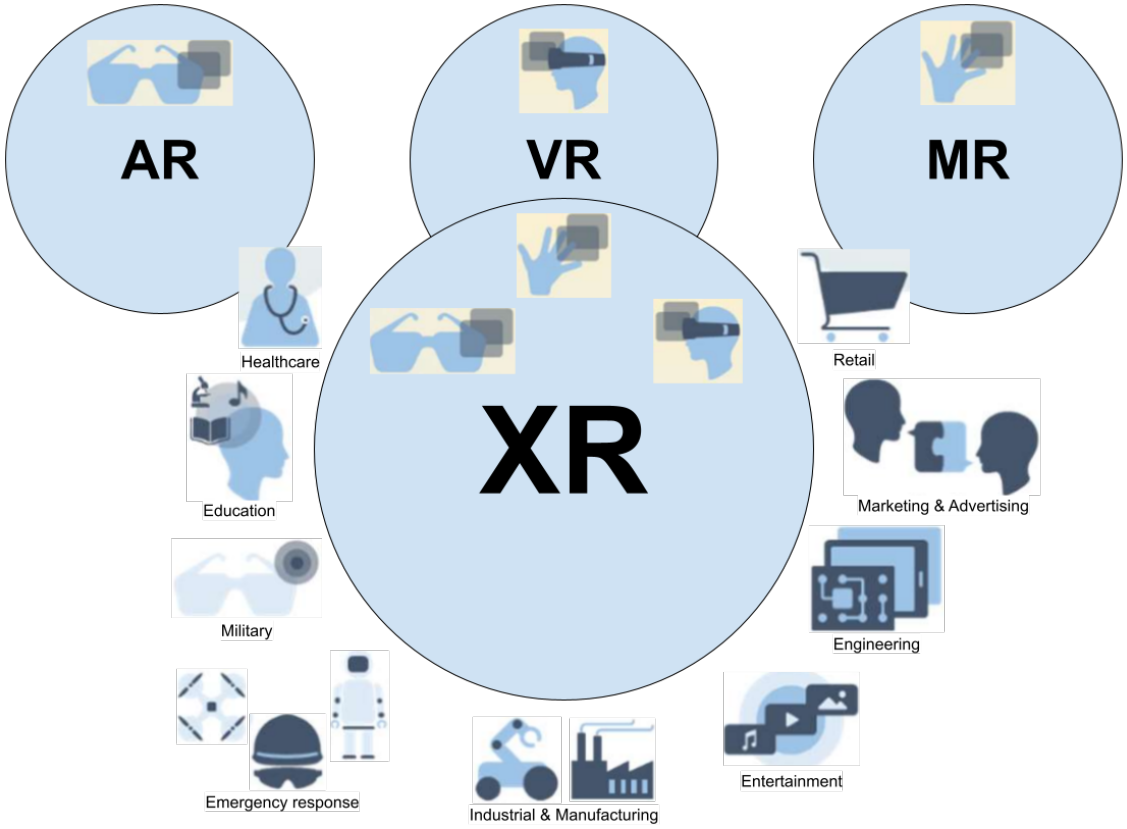

Figura 2.7: *Tipos e diferenças de realidades*

usuário, como ilustrado na Figura 2.7. De fato, o 3GPP classifica os casos de uso de XR em três categorias relacionadas, a saber: VR, AR e jogos em nuvem (*cloud gaming*) [3GPP 2021-d].

A VR gera um mundo virtual onde o usuário está totalmente imerso, criando assim uma sensação de presença física que transcende o mundo real. Serviços de VR geralmente são habilitados por *streaming* dependente do campo de visão do usuário, ou seja, um esquema de *streaming* adaptativo que ajusta a taxa de bits do vídeo 3D usando o status da rede e as informações de pose do usuário. Especificamente, a cena 3D omni-direcional em relação à posição do observador é dividida espacialmente em sub-imagens ou *tiles* independentes. O servidor de *streaming* oferece várias representações do mesmo *tile*, armazenando-o em diferentes qualidades. A transmissão de novo conteúdo pode ser acionada pelos movimentos do usuário, bem como pela demanda da próxima parte do vídeo. Após o download de todos os *tiles* do campo de visão, eles são renderizados, gerando a representação 3D exibida ao usuário. Os serviços de VR apresentam vídeo de alta taxa em *downlink* complementado por atualizações frequentes de pose em *uplink*.

A AR combina objetos virtuais com a visão 3D ao vivo do mundo real, criando assim um ambiente personalizado e realista, com o qual o usuário pode interagir. Portanto, estimar a localização do usuário e o seu campo de visão também são relevantes em serviços de AR. No entanto, diferentemente das soluções de VR, as soluções de AR

normalmente não dependem de sensores de detecção de movimento caros, mas os complementam com câmeras montadas em óculos AR. Portanto, serviços AR são frequentemente apresentados por um fluxo de vídeo em *uplink*. O vídeo é transmitido continuamente para o servidor XR que realiza rastreamento de pose para estimar a posição e orientação do usuário via localização e mapeamento simultâneos (*simultaneous localization and mapping* - SLAM). O campo de visão do usuário então estimado é usado para gerar uma cena 3D aumentada onde objetos virtuais são sobrepostos em certas posições. Por fim, os objetos 3D ou vídeo renderizados são codificados e transmitidos de volta para o UE. Um vídeo em *uplink* de qualidade média que capture os objetos principais é suficiente para o SLAM. Portanto, com o objetivo de reduzir a taxa de bits em *uplink*, a qualidade do fluxo de vídeo nesta direção é frequentemente reduzida em comparação com o fluxo em *downlink*.

Jogos em nuvem refere-se a uma aplicação de jogo interativo executado no servidor de nuvem. Dessa forma, tarefas computacionalmente intensivas são descarregadas do dispositivo para a nuvem, aliviando o consumo de recursos e o consumo de energia no UE. Em um cenário típico, o servidor gera uma sequência de cenas 2D ou 3D como um fluxo de vídeo em resposta a um comando de controle enviado pelo UE. Para um serviço de jogos em nuvem, os sinais de controle incluem entradas do controlador portátil e amostras de movimento em 3 ou 6 graus de liberdade (3DoF/6DoF). 3DoF refere-se aos dados de rotação, enquanto 6DoF também adiciona as informações sobre o deslocamento do UE nas dimensões X, Y e Z. Vários jogadores podem participar da mesma sessão de jogo. No entanto, é importante observar que o fluxo de vídeo resultante em jogos em nuvem depende das ações do usuário. Portanto, similar a um serviço de VR, atualizações frequentes de movimento/controle são necessárias em *uplink*.

Neste trabalho, nosso foco é em um estudo de caso envolvendo AR, mais especificamente *Mobile Augmented Reality* (MAR), ou seja, aplicações AR consumidas em dispositivos móveis, como *smartphones* e óculos de VR/AR (denominados *Head-Mounted Displays* - HMD). Dessa forma, na seção a seguir, discutimos a importância das tecnologias 5G e computação de borda para a realização de aplicações MAR.

2.4 Arquitetura 5G, computação de borda e MAR

Aplicações MAR criam oportunidades para novas formas de interação, aprendizado ou entretenimento entre humanos. Na indústria, por exemplo, os dispositivos móveis de AR podem exibir instruções em tempo real no local para um operador ou detectar automaticamente falhas de segurança ou procedimentos incorretos [Cao et al. 2023].

Para viabilizar completamente os casos de uso relacionados a MAR, as aplicações não devem apenas posicionar adequadamente o conteúdo virtual, mas precisam

também permitir que tanto o usuário quanto o cenário real interajam com esse conteúdo. Isso exige que o ambiente real seja analisado com a maior precisão possível e em tempo real, requerendo vários algoritmos pesados, como segmentação semântica e reconstrução 3D, para serem executados simultaneamente em tempo real. Embora existam algumas implementações de última geração em tempo real desses algoritmos, eles exigem hardware com grande capacidade de processamento, que geralmente não são encontrados em dispositivos portáteis. Além disso, esses algoritmos consome muita energia, a qual também é limitada em dispositivos móveis. Outra dificuldade de MAR é a latência entre uma ação no mundo real e sua respectiva representação renderizada e apresentada na tela, a qual deve ser inferior a 20 ms para uma experiência ideal e não mais que 60 ms para um serviço aceitável [Morín, Pérez e Armada 2022]. Essa latência estrita, juntamente com requisitos de hardware muito exigentes, justificam a ideia de descarregar alguns ou todos esses algoritmos computacionalmente intensivo do dispositivo (UE).

Nesse contexto, as redes 5G e a computação de borda tornam-se essenciais para a completa realização das aplicações MAR. As redes 5G, com suas capacidades avançadas de alta largura de banda, baixa latência e suporte para inúmeros dispositivos conectados, oferece a infraestrutura necessária para a implantação de MAR em alta escala. A computação de borda, por sua vez, complementa essa infraestrutura ao aproximar o processamento e o armazenamento de dados dos usuários finais, garantindo latências menores. Quando combinadas, essas tecnologias criam um ecossistema escalável e dinâmico, capaz de suportar aplicações que exigem tanto desempenho quanto confiabilidade.

Em resumo, a integração da arquitetura 5G, computação de borda e XR representa um avanço significativo na evolução das redes e aplicações digitais. Essa combinação não apenas permite a criação de novas experiências imersivas e interativas, mas também redefine como as redes e os serviços digitais são estruturados e operados. Ao eliminar as barreiras tradicionais de latência e capacidade de processamento, essas tecnologias em conjunto possibilitam a realização de aplicações que antes eram consideradas impossíveis, abrindo novas oportunidades para inovação em diversas áreas.

2.5 Trabalhos relacionados

Os trabalhos relacionados ao tema da pesquisa deste estudo podem ser classificados em dois grupos principais. O primeiro grupo compreende os trabalhos relacionados à implementação e o uso da arquitetura do sistema 5G, com foco na indústrias verticais e arquiteturas de APIs *Northbound*. Nenhum trabalho neste grupo discute especificamente a implantação de uma aplicação MAR. O segundo grupo abrange os trabalhos que investigam a implementação e implantação de aplicações MAR em redes de borda. Os trabalhos

no segundo grupo, no entanto, não se preocupam com a integração das aplicações com redes 5G. A seguir, discutimos trabalhos relevantes em cada grupo.

No primeiro grupo, o trabalho em [Tangudu et al. 2020] aborda a exposição de serviços de rede por meio do CAPIF para fornecer consistência e facilitar o desenvolvimento de aplicações de terceiros que usam as APIs 3GPP *Northbound*. Esse trabalho é estendido posteriormente em [Fragkos et al. 2021], onde os autores discutem como o 5G permite que indústrias verticais utilizem todos os recursos da rede, detalhando os arcabouços CAPIF e SEAL. Os autores também apresentam alternativas para implementar tais arcabouços em tecnologias nativas de nuvem. Os autores em [Sanchez et al. 2022] evoluem essa discussão implementando o componente *CAPIF Core Function* como uma entidade central, visando estabelecer uma base para um ecossistema onde são disponibilizados recursos de integração com a rede de núcleo. Mais recentemente, os autores em [Charismiadis et al. 2023] desenvolveram o arcabouço CAPIF e disponibilizaram o a implementação como código aberto. Como uma prova de conceito, os autores aplicaram os serviços CAPIF em um sistema de gerenciamento de eventos.

No segundo grupo, os autores em [Toczé e et al. 2019] introduzem o MR-Leo, um protótipo que visa melhorar o streaming de vídeo no contexto de MAR. O MR-Leo descarrega a criação da nuvem de pontos e a renderização gráfica do dispositivo para a borda da rede. O desempenho do protótipo é analisado em relação à latência e taxa de transferência em diferentes configurações, considerando alternativas para o protocolo de transporte, o formato de compressão de vídeo e os dispositivos utilizados. Um arcabouço MAR abrangente, denominado ARENA, foi apresentado em [Pereira et al. 2021], com o objetivo de simplificar a construção e hospedagem de aplicações de AR e VR voltadas para navegadores. O ARENA fornece vários componentes, incluindo: um serviço de diretório geoespacial hierárquico que conecta usuários a servidores e conteúdo próximos, um sistema de autenticação baseado em *token* para controlar o acesso do usuário ao conteúdo e um barramento PubSub que transmite todas as interações dos usuários em tempo real. Apesar dos recursos disponíveis, o ARENA não foi avaliado em relação à descarga das tarefas MAR para a borda, mas sim, em relação ao posicionamento do *broker* de mensagem na borda da rede e na nuvem.

A Tabela 2.1 resume as principais características dos trabalhos discutidos acima. Como podemos notar, diferentemente da nossa proposta, nenhum trabalho aborda, de forma conjunta, a entrega de aplicações XR, computação de borda e rede 5G.

Artigo	Computação de borda	XR	5G
[Fragkos et al. 2021]	✗	✗	✓
[Sanchez et al. 2022]	✗	✗	✓
[Tangudu et al. 2020]	✗	✗	✓
[Charismiadis et al. 2023]	✗	✗	✓
[Toczé e et al. 2019]	✓	✓	✗
[Pereira et al. 2021]	✓	✓	✗
[Hammad et al. 2023]	✓	✓	✗
Este trabalho	✓	✓	✓

Tabela 2.1: *Artigos relacionados e suas principais características*

2.6 Considerações finais

As redes 5G, juntamente com a computação de borda, representam um salto evolutivo na forma como interagimos com o mundo, diluindo as fronteiras entre o real e o virtual. Este avanço tecnológico redefine não só a interação humana com o universo digital, mas também expande as possibilidades de consumo e interatividade além dos dispositivos IoT convencionais e *smartphones*. No entanto, a implementação prática de aplicações XR sobre essas tecnologias ainda enfrenta desafios devido, em grande parte, à quantidade de software envolvida e à complexidade da integração. As redes 5G, fundamentada na SBA, proporciona uma infraestrutura flexível para o desenvolvimento e gerenciamento de serviços. Porém a eficácia da integração das aplicações depende da facilidade de uso pelos desenvolvedores. Arcabouços como o CAPIF e o SEAL, apesar de essenciais para padronizar o acesso de aplicações de terceiros a serviços da rede, ainda apresentam muita complexidade de uso e manutenção, o que inviabilizou, até onde sabemos, a aplicação desses arcabouços pela industria de jogos e outras aplicações que fazem uso de imersão. No próximo capítulo, apresentaremos um modelo de *streaming* de mídia para o 5G definido pelo 3GPP para preencher essa lacuna.

CAPÍTULO 3

Entrega de um Serviço MAR numa Rede 5G

Neste capítulo, apresentamos uma proposta para entregar um serviço MAR em uma rede 5G usando computação de borda. Inicialmente, apresentamos como o 3GPP especificou a entrega de serviços XR dentro da arquitetura de redes 5G, denominado *5G Media Streaming* (5GMS) (Seção 3.1). Em seguida, descrevemos com mais detalhes o protótipo XR utilizado neste trabalho (Seção 3.2). Na seção seguinte (Seção 3.3) comparamos qualitativamente o protótipo utilizado com o modelo 5GMS. Por fim, na Seção 3.4 realizamos a avaliação quantitativa do protótipo em experimentos com diversos cenários.

3.1 XR no 3GPP

Com o avanço das redes móveis, dispositivos com maior capacidade de processamento e a necessidade de aplicações cada vez mais imersivas, a integração de serviços XR com aplicações móveis tem se tornado cada vez mais relevante. Dada a complexidade de se trabalhar com o CAPIF e o SEAL, para atender essa demanda, o 3GPP especificou dois métodos simplificados para o fornecimento de serviços de *streaming* de mídia e XR, a saber: *Explicit Congestion Notification Low Latency, Low Loss and Scalable Throughput* (L4S) e o modelo de *streaming* de mídia 5G (5GMS) [3GPP 2023]. Este último é o modelo adotado neste trabalho, sendo descrito a seguir.

O 5GMS assume que um UE executa a parte cliente de uma aplicação que deseja utilizar uma rede 5G para acessar serviços XR na rede de dados (*data network* - DN). Essa aplicação deverá interagir com as funções de rede do 5G, bem como com uma camada de serviços XR. A Figura 3.1 ilustra os principais componentes dessa arquitetura. Componentes em amarelo fazem parte do 5GMS. Componentes em cinza representam funções/elementos da rede 5G. Componentes em azul estão associados a aplicações de terceiros (aplicação cliente/servidor que faz uso de serviços XR).

O modelo 5GMS é uma extensão da arquitetura 5G cujo objetivo é suportar um conjunto de serviços comumente utilizados em aplicações XR. Os principais módulos que compõem essa camada de serviço são: *5GMS Application Function* (5GMS AF),

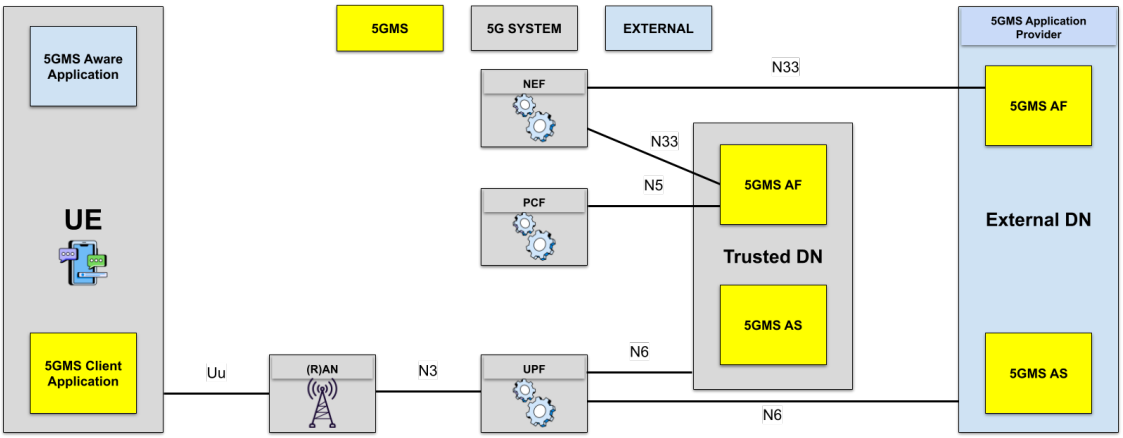

Figura 3.1: *Arquitetura geral do modelo 5GMS*

5GMS Application Server (5GMS AS) e *5GMS Client Application*. O componente *5GMS Client Application* é um módulo que executa no UE e que é acessado pela *5GMS Aware Application* via interfaces bem definidas. Esse módulo se comunica com o 5GMS AF para estabelecer, controlar e transmitir/receber dados de uma sessão XR. O componente 5GMS AS é um servidor de aplicação que implementa serviços XR, incluindo suporte nativo a *streaming* de vídeo de alta qualidade, rastreamento de alta precisão para posição e movimento, renderização e interação em tempo real. O componente 5GMS AF é uma AF escrita especificamente para serviços XR. Essa função pode influenciar o roteamento de dados da aplicação, interagir com a PCF para controlar políticas específicas para XR e interagir com a NEF para consumir serviços expostos pelo núcleo 5G.

Diferentemente do componente *5GMS Client Application* que executa no UE, os componentes 5GMS AS e 5GMS AF executam em uma rede de dados (DN) e se comunicam com o UE através das interfaces N3, N6 e Uu (interface entre o UE e a RAN). A DN onde esses componentes executam pode ser considerada confiável ou não confiável (externa). Quando esses componentes executam em uma DN confiável, eles podem se comunicar diretamente com todas as funções do núcleo 5G. No entanto, quando esses módulos executam em DNs externas, eles podem se comunicar com o núcleo apenas através da NEF, usando a interface N33.

A Figura 3.2 apresenta detalhes do *5GMS Client Application*. Internamente, esse componente é constituído pelas funções *Media Session Handler* e Media Stream Handler. Essas funções podem ser invocadas pelo lado cliente da aplicação (*5GMS Aware Application*) por meio das APIs M6 e M7, assim como as aPIs também podem ser usadas para comunicação das funções entre si. inclusive expondo as APIs para fora do 5GMS Client.

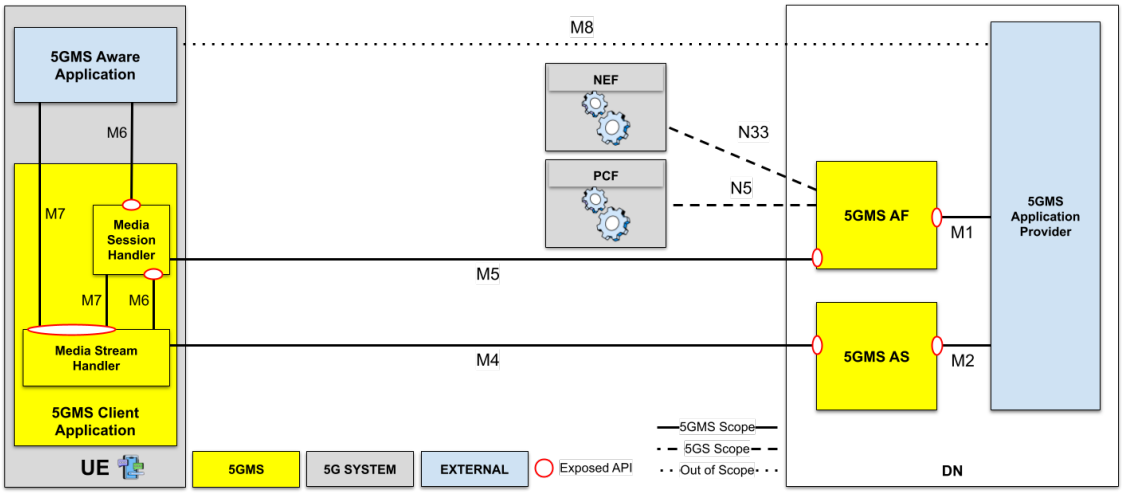

Figura 3.2: *Arquitetura detalhada do modelo 5GMS*

3.2 MR-Leo

O aumento da demanda por aplicações XR tem evidenciado limitações significativas em termos de processamento e consumo de energia. Embora plataformas como Google ARCore [Google 2024] e Apple ARkit [Apple 2024] tenham fomentado o desenvolvimento de aplicações XR, a capacidade de processamento dessas aplicações ainda é limitada pelos recursos de hardware dos UEs, resultando em problemas como alta latência e rápida drenagem da bateria. Nesse contexto, como discutido no Capítulo 2, a computação de borda surge como uma solução promissora ao transferir o processamento intensivo para servidores localizados próximos aos UEs, aliviando a carga sobre os dispoistivos e permitindo uma experiência de XR mais fluida e eficiente.

O *Mixed Reality Linköping Edge Offloading* (MR-Leo) [Toczé e et al. 2019] é um protótipo desenvolvido na Universidade de Linköping que explora a utilização da computação de borda para habilitar serviços de realidade mista, particularmente AR, em dispositivos móveis.

O MR-Leo cria streaming de vídeos e inclui elementos virtuais às cenas capturadas, possibilitando o *offloanding* de algumas tarefas, como criação de nuvem de pontos e renderização do vídeo na borda da rede. As principais funções executadas pelo protótipo incluem: (1) a captura do mundo real, geralmente por meio de um *streaming* de vídeo; (2) a análise do vídeo capturado para a construção de uma representação virtual do mundo real, denominada nuvem de pontos; (3) criação e adição de objetos virtuais em pontos específicos do vídeo capturado; e (4) envio do vídeo enriquecido para o usuário.

A Figura 3.3 apresenta a arquitetura do MR-Leo, a qual segue o modelo cliente/servidor. O lado cliente, executa no dispositivo do usuário, sendo responsável apenas pelas tarefas de captura e exibição do vídeo enriquecido. O cliente MR-Leo aceita dois tipos de entrada: o vídeo capturado pela câmera e os comandos de inserção de objetos

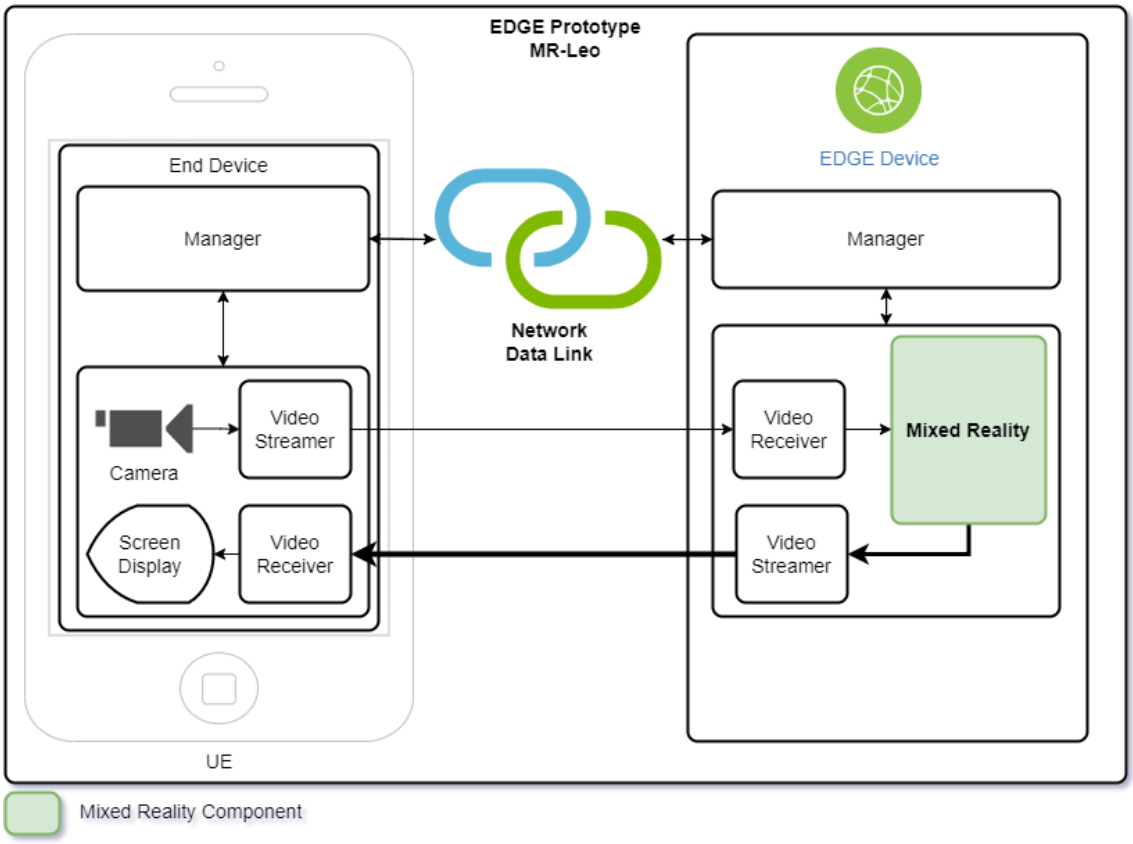

Figura 3.3: *Arquitetura básica do protótipo MR-Leo*

virtuais. Como saída, exibe tanto o vídeo capturado originalmente, quanto o vídeo enriquecido pelo processamento no servidor. No lado cliente, a transmissão e recepção do vídeo são realizadas pelos componentes *video streamer* e *video receiver*, respectivamente.

No lado servidor, no servidor de borda (*edge device*), duas entradas são esperadas: o vídeo capturado pela câmera e enviado pelo cliente e as mensagens de comando para inserção dos objetos virtuais. O componente central do lado servidor é o módulo *Mixed Reality*, onde as tarefas de criação da nuvem de pontos e criação e adição dos objetos virtuais são executadas. Essas tarefas são as mais demandantes em termos de processamento. Os componentes *Video Receiver* e *Video Transmitter* no lado servidor são responsáveis, respectivamente, por receber o vídeo enviado e transmitir o vídeo enriquecido com objetos virtuais para o cliente.

Conforme mostrado na Figura 3.3, tanto o cliente quanto o servidor possuem um componente *Manager* responsável por configurar os diferentes componentes e lidar com a comunicação entre eles. É importante saber que antes de iniciar uma sessão, os *Managers* precisam iniciar um processo de negociação e sincronização definindo a sessão de transmissão e recebimento de vídeo. Após esse processo, a sessão é iniciada e o lado cliente inicia a captura do vídeo pela câmera do UE em tempo real e encaminha o *streaming* para o componente *Video Receiver* do servidor. Este recebe o fluxo que

é encaminhado para o componente *Mixed Reality*. Esse, por sua vez, cria uma nuvem de pontos tridimensional mapeando o ambiente capturado, determinando a posição e a rotação do dispositivo do usuário em relação ao quadro do vídeo. Esta nuvem é atualizada conforme os próximos quadros são recebidos para que o posicionamento seja atualizado. Quando uma nuvem de pontos é criada e o posicionamento do UE é conhecido, os objetos gráficos podem ser adicionados com base nos pontos anteriormente mapeados. O momento de adicionar um objeto à imagem é uma decisão do usuário por meio da interface da aplicação. Uma vez que o vídeo está com os elementos incluídos, ele é encaminhado ao componente *Video Transmitter* do servidor, que o reencaminha para o componente *Video Receiver* do cliente. Finalmente, o cliente apresenta o vídeo enriquecido e o vídeo capturado originalmente para o usuário por meio do componente *Screen Display*.

A parte cliente do MR-Leo foi implementada para funcionar, inicialmente, em dispositivos com sistema operacional Android. Já a parte servidor foi projetada para executar em uma máquina com arquitetura X86-64 com sistema operacional Linux. O parte servidor é implementada em C++. No protótipo também foram utilizados os softwares de código aberto: (1) Gstreamer [Gstreamer 2024], para implementar as partes de transmissão/recebimento de vídeo; (2) ORB-SLAM2 [Raul Mur-Artal, Juan D. Tardos, J. M. M. Montiel and Dorian Galvez-Lopez (DBoW2) 2024], para técnicas de SLAM; e (3) Pangolin [Pangolin 2024], para renderizar figuras 3D.

Para avaliar o desempenho, os autores do MR-Leo definiram métricas onde foram observados os aspectos de latência e taxa de transferência, ambos fundamentais para garantir, minimamente, uma experiência aceitável de XR. A latência foi medida de duas maneiras, a primeira, denominada "tempo até o elemento virtual"(*Time To Virtual Element* - T2VE), que avalia o tempo entre a interação do usuário (como pressionar um botão para adicionar um elemento virtual) e a exibição desse elemento na tela. A segunda, "tempo de ida e volta do quadro"(*Frame Round Trip Time* - FRTT), mede o tempo necessário para que um quadro de vídeo capturado seja processado no servidor de borda e retornado ao UE com a sobreposição XR. A taxa de transferência, por sua vez, mede a quantidade de quadros processados com sucesso pela borda e exibidos no UE, sendo um indicador direto de quão bem os elementos virtuais se integram à realidade. Essas métricas são usadas para avaliar o *QoS* e a responsividade da aplicação XR em diferentes configurações de rede e processamento.

3.3 Comparação qualitativa do MR-Leo e o modelo 5GMS

O modelo 5GMS foi projetado para oferecer serviços de *streaming* de mídia na rede 5G. Destacamos três pontos nesse contexto: (1) Escalabilidade e suporte a múltiplos dispositivos conectados; (2) O desenvolvimento da Service-Based Architecture (SBA) e interfaces padronizadas; e (3) baixa latência e alta confiabilidade (URLLC).

O MR-Leo foi desenvolvido como um protótipo para demonstrar o potencial da computação de borda, mitigando as limitações de hardware dos dispositivos móveis. Em comparação, destacamos três aspectos do protótipo: (1) Offloading de carga de trabalho computacionalmente intensiva para a borda da rede; (2) Arquitetura modular; (3) flexibilidade para trabalhar com diferentes soluções de codificação de video e protocolos de transporte.

Ao compararmos as Figuras 3.1, 3.2 e 3.3 observamos pontos de convergência entre o modelo 5GMS e o MR-Leo. Por exemplo, em ambas as soluções, notamos um componente cliente/servidor responsável por negociar as configurações (e.g., codificadores/decodificadores, protocolo de rede, etc.) usada na seção do usuário. Esse componente é representado pelo Manager no MR-Leo e *5GMS Aware Application* no 5GMS. Como o componente *5GMS Client Application*, a parte cliente do MR-Leo possui um conjunto de classes para tratar a seção do usuário e o *streaming* de mídia. Esse último é oferecido pelo Gstreamer. No lado servidor do MR-Leo, o componente *Mixed Reality* é funcionalmente equivalente ao componente 5GMS AS do modelo do 3GPP. *Mixed Reality* oferece suporte nativo a *streaming* de vídeo de usando o Gstreamer, rastreamento de alta precisão para posição e movimento usando ORB-SLAM2 e renderização em tempo real pelo uso do Pangolin.

Contudo, diferentemente do modelo 5GMS, o MR-Leo não possui um componente funcionalmente equivalente ao 5GMS AF. Isso significa que o protótipo não possui nativamente formas de interagir com os serviços do núcleo 5G para, por exemplo, influenciar políticas de QoS. Isso é esperado, uma vez que o protótipo foi concebido apenas dentro do contexto de computação de borda, onde o dispositivo se conecta com a borda da rede via WiFi.

Mesmo sem o suporte nativo à tecnologia 5G, na próxima seção, apresentamos e avaliamos algumas soluções para integrar o MR-Leo a uma rede 5G.

3.4 Avaliação quantitativa envolvendo o MR-Leo e uma Rede 5G

Nesta seção, apresentamos uma avaliação envolvendo o uso de uma arquitetura de rede móvel 5G e uma aplicação XR com offloading em um servidor de borda, conforme descrito a seguir.

3.4.1 Experimento inicial e base de comparação

Com a intenção de obtermos uma base de comparação, decidimos refazer o experimento realizado com o protótipo de aplicação XR conforme descrito em [Toczé e et al. 2020]. Ao iniciarmos, logo percebemos que seria preciso adaptações na execução. Para a execução deste experimento não foi possível utilizar os mesmos recursos de hardware e software devido à evolução tecnológica no lapso temporal, uma vez que a execução do experimento aconteceu em 2024. Considerando a evolução tecnológica, mantemos a topologia de rede cliente/servidor, originalmente proposta e disposta com um dispositivo cliente (um *smartphone*), representando o UE, um dispositivo servidor, representando o servidor de borda e um roteador desconectado da Internet, representando o ponto central de ancoragem.

O servidor de borda possui as seguintes especificações: um laptop Lenovo Thinkpad E14 Gen 4, modelo 21E4 fabricado em 27/06/2023, 40 GB DDR4 3200 MHz de RAM, um processador Intel Core i7-1255U (1.70 GHz, 10 núcleos, 12 threads), GPU Iris Xe Graphics Eligible ADL GT2 4 GB VRAM LPDDR4X-4266, SSD 512 GB M2 PCIe, Ethernet Gigabit Intel I219 e Intel Wifi 6 802.11ax AX201 160 Mhz 2.4 Gbps. No servidor, nenhuma outra aplicação estava em execução em segundo plano. Substituímos o Sistema Operacional (SO) originalmente instalado por um SO Linux Ubuntu Desktop 20.04.

Como UE, usamos um *smartphone* Samsung Galaxy A14 5G, display 6,6"2400x1080 pixels 90 Hz, processador Exynos 1330 Octa-core (Cortex A78x2 2.4GHz + Cortex A55x6 2.0GHz), 4 GB RAM, armazenamento interno 128 GB, GPU Mali G68 MP2, câmera traseira 50 MP, câmeras adicionais 2 MP para desfoque de cenário e 2 MP, câmera frontal 13 MP, bateria 5.000 mAh, modem 5G NR Sub-6GHz 2.55Gbps (DL)/1.28Gbps (UL) LTE Cat.18 6CC (DL)/Cat.18 2CC (UL) e conectividade Wi-Fi 802.11ac (2x2 MIMO) Dual-band, Bluetooth 5.2, FM Radio Rx. No UE todos os aplicativos de terceiros foram desinstalados ou desabilitados.

Para o ponto de ancoragem, usamos um roteador Mercusys AC12G(EU), 4 antenas fixas omnidirecionais, 300 Mbps em 2.4 GHz 802.11b/g/n e 867 Mbps em 5 GHz 802.11a/n/ac, conectividade 1 x Porta Gigabit WAN e 3 x Porta Gigabit LAN.

Topologicamente os componentes da rede foram dispostos da seguinte forma. O roteador foi conectado ao servidor de borda por meio de cabo UTP Gigabit Ethernet (1000 Mbit/seg) e o cliente UE conectado pela interface de rede sem fio 802.11ac. A topologia usada neste experimento está ilustrada na Figura 3.4.

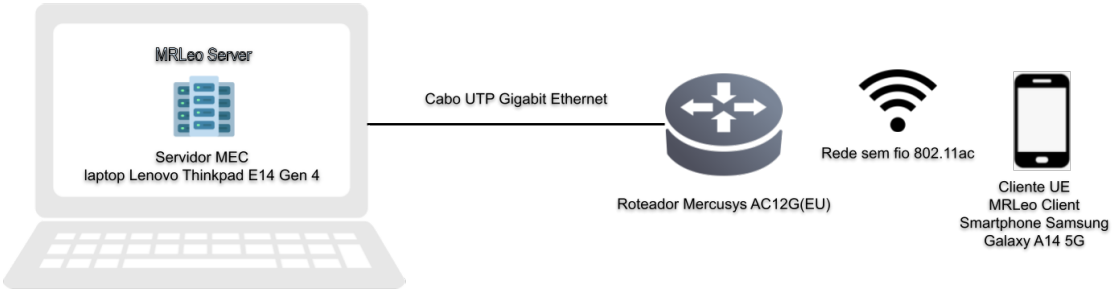

Figura 3.4: *Topologia do experimento que serve como base de comparação*

As configurações do MR-Leo utilizadas no experimento consistem em diferentes combinações de protocolos de transporte e formatos de compressão de vídeo e são apresentadas na Tabela 3.1. Para analisar o impacto desses elementos no desempenho da aplicação XR, foram implementadas variações utilizando o codec H.264, que é amplamente adotado por sua eficiência de compressão, em conjunto com os protocolos de rede TCP e UDP. O uso de TCP visa garantir a entrega confiável de pacotes, embora possa causar aumento na latência, enquanto o UDP, por sua vez, sacrifica a confiabilidade em prol de menor atraso na transmissão.

O experimento foi conduzido em um ambiente de rede local controlado, onde os dispositivos estavam posicionados a uma curta distância do ponto de ancoragem, assegurando uma conexão estável e minimizando interferências externas. Para garantir a consistência dos testes nos cenários, foi utilizado um vídeo pré-gravado de 60 segundos, com resolução de 640x480 pixels e taxa de 30 quadros por segundo (fps), em vez de capturas de imagens em tempo real [Klervie Toczé 2019]. A inserção de objetos virtuais na cena foi automatizada, ocorrendo em intervalos de 10 segundos.

Na tabela 3.1 apresentamos os cenários e suas variações de compressão de vídeo e protocolos de transporte utilizados no experimento: (1) a primeira coluna apresenta o identificador do cenário; (2) a segunda coluna traz o protocolo de transporte na rede e o padrão de compressão de vídeo utilizado no servidor de borda; (3) a terceira coluna traz o protocolo de transporte na rede e o padrão de compressão de vídeo utilizado no cliente UE. Neste último, ainda complementamos as informações do tipo de codificador e a biblioteca de *streaming* utilizada. Para cada cenário, foram executados 30 experimentos e em cada experimento foram coletadas a latência fim-a-fim (em ms) e a vazão (*throughput*) (em fps).

Cenários dos Experimentos	Servidor de borda		Cliente UE			
	PTR	**PCV**	**PTR**	**PCV**	**COD**	**STR**
1	TCP	H.264	TCP	H.264	SW	GS
2	UDP	H.264	UDP	H.264	HW	RL
3	UDP	H.264	UDP	H.264	SW	GS

Legenda:

PTR:	Protocolo de Transporte na Rede
PCV:	Padrão de Compressão de Vídeo
COD:	Codificador de Software (SW) ou de Hardware (HW)
STR:	Streaming com RL ou GS
RL:	RtpLib - Comunicação de rede em tempo real
GS:	GStreamer - Processamento e pipeline de mídia

Tabela 3.1: *Cenários dos experimentos*

Cenários dos Experimentos	End-to-End (E2E) (ms)			Throughput (fps)		
	M	**Mo**	**DP**	**M**	**Mo**	**DP**
1	37,0000	33,0000	25,0197	18,0000	30,0000	8,8321
2	34,0000	33,0000	4,3647	29,0000	29,0000	8,8288
3	33,0000	33,0000	9,0881	30,0000	30,0000	9,7878

Legenda:

M:	Média
Mo:	Moda
DP:	Desvio Padrão

Tabela 3.2: *Resultados do experimento que serve de base de comparação*

A Tabela 3.2 apresenta os resultados, considerando as métricas de média (M), moda (Mo) e desvio padrão (DP). Esses resutados também são apresentados nas Figuras 3.5 e 3.6. Como mostrado na Figura 3.5, no cenário 1, latência média foi de 37 ms, com uma moda de 33 ms e um desvio padrão relativamente alto, de 25,0197 ms, indicando variabilidade nos tempos de resposta. No cenário 2, houve uma redução na latência média para 34 ms, mantendo a moda em 33 ms, enquanto o desvio padrão diminuiu drasticamente para 4,3647 ms, destacando-se como um cenário com maior estabilidade na entrega de pacotes. Já no cenário 3, a média foi ainda menor, de 33 ms, com moda também em 33 ms e desvio padrão de apenas 9,0881 ms.

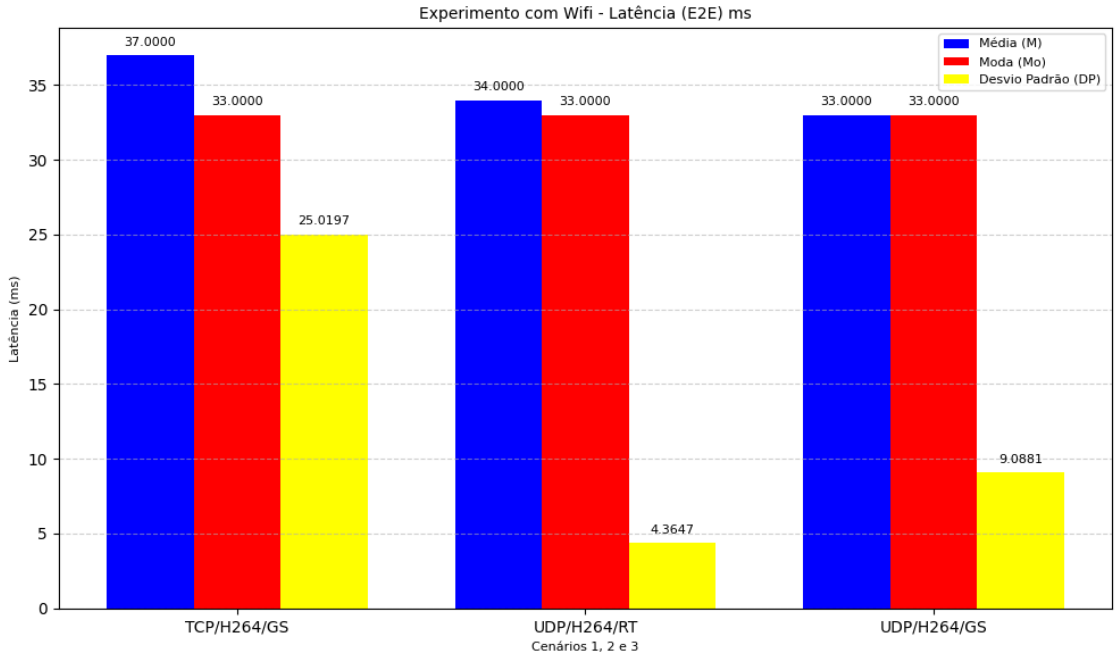

Figura 3.5: ***Latência (E2E) dos cenários 1, 2 e 3 para o experimento que serve como base de comparação***

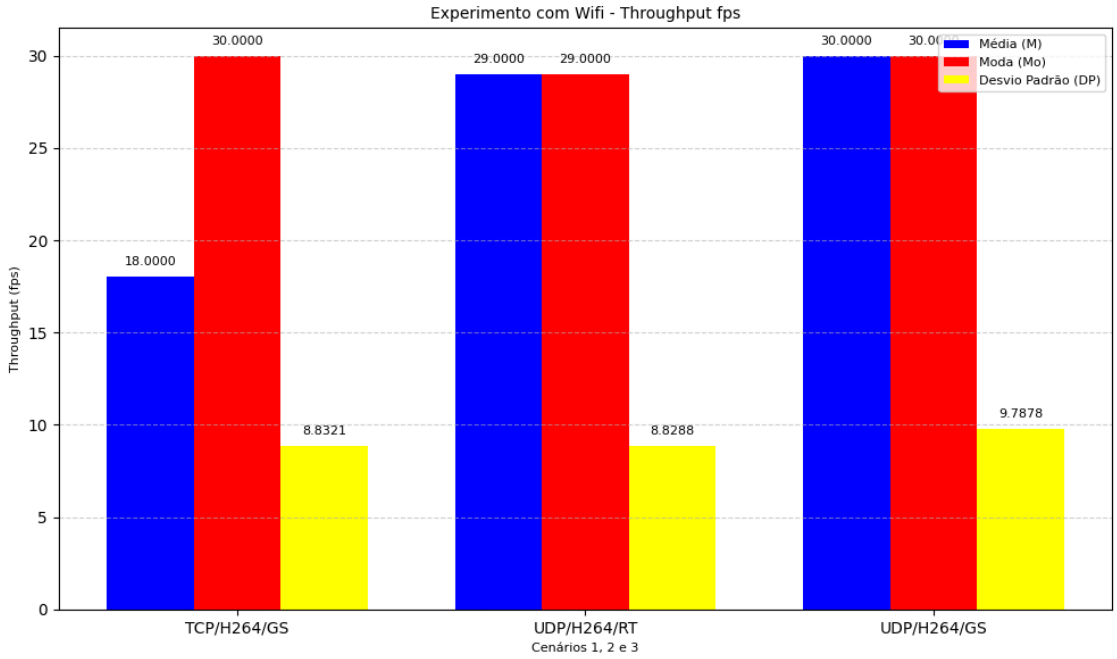

Figura 3.6: ***Throughput dos cenários 1, 2 e 3 para o experimento que serve como base de comparação***

A Figura 3.6 apresenta os resultados de vazão. No cenário 1, observou-se uma média de latência mais alta, de 37 ms, com throughput médio significativamente menor, de 18 fps. No cenário 2, houve uma melhoria perceptível em ambas as métricas, a média de latência reduziu para 34 ms, enquanto o throughput médio aumentou para

29 fps. A moda para ambas as métricas permaneceu constante em 33 ms e 29 fps, respectivamente, indicando consistência nos valores. O desvio padrão da latência caiu para 4,3647 ms, enquanto o do throughput manteve-se relativamente baixo em 8,8288 fps, evidenciando maior previsibilidade do desempenho sob UDP. O cenário 3 apresentou resultado ligeiramente melhor em relação à latência e throughput, com uma média de latência de 33 ms e throughput médio de 30 fps. Ambas as métricas apresentaram moda de 33 ms e 30 fps, respectivamente. Os resultados do experimento de base de comparação mostram que o protocolo UDP, nos cenários 2 e 3, apresentam vantagens de desempenho, enquanto o TCP, no cenário 1, apresenta maior variabilidade e throughput limitado.

3.4.2 Experimento com emuladores

Neste experimento, implantamos o MR-Leo em uma rede 5G emulada. Para construir o ambiente emulado decidimos implantar o Free5GC na versão 3.3.0 [free5GC 2024] que é um projeto de código aberto de licença Apache 2.0. O código está disponível no repositório do GitHub [free5GC 2024]. Para simular a RAM, utilizamos o projeto o UERANSIM versão 3.2.6 [UERANSIM 2024] que é um simulador também de código aberto para UEs 5G e estação de rádio base (gNodeB). O código está disponível no repositório do GitHub [UERANSIM 2024]. Ambos os projetos são amplamente utilizados pela comunidade científica para experimentos.

Para este experimento, utilizamos os mesmos equipamentos da sessão 4.1, adicionando alguns elementos na rede conforme a topologia apresentada na Figura 3.7. No experimento, a rede 5G foi configurada utilizando o UERANSIM e o free5GC para emular a comunicação entre o UE e o 5GC. O UE estabelece a conexão com a gNodeB por meio de uma interface virtual criada pelo simulador UERANSIM, denominada uesimtun0, que cria um túnel de comunicação para transportar os pacotes de dados até o 5GC. UERANSIM executa na máquina hospedeira (notebook), enquanto o free5GC executa em uma máquina virtual (VM) na máquina hospedeira. A interface enp0s8 da VM é responsável por intermediar a comunicação entre a gNodeB e o 5GC, que autentica e gerencia as sessões do UE, conectando-o à DN, representada pelo servidor de borda. Esse servidor é configurado com a parte servidor do MR-Leo. A parte cliente do MR-Leo é executada pelo UE (*smartphone*). O fluxo de dados segue um caminho bidirecional: o UE envia requisições encaminhadas pela gNodeB e pelo 5GC até o servidor de borda na DN, e as respostas seguem o caminho inverso, permitindo a avaliação do desempenho e da latência na comunicação entre os componentes da rede emulada.

A Figura 3.8 apresenta um diagrama de sequência das principais mensagens trocadas entre o UE, a gnodeB, as funções do core e a DN desde a autenticação do UE até a criação e estabelecimento de uma sessão de dados entre as partes cliente e servidora do

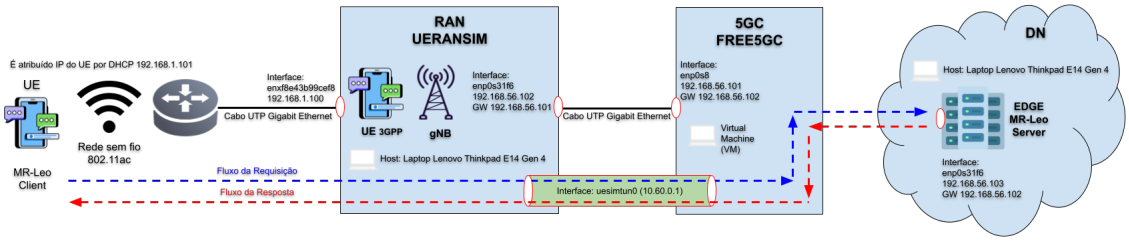

Figura 3.7: *Topologia do experimento com emuladores*

MR-Leo.

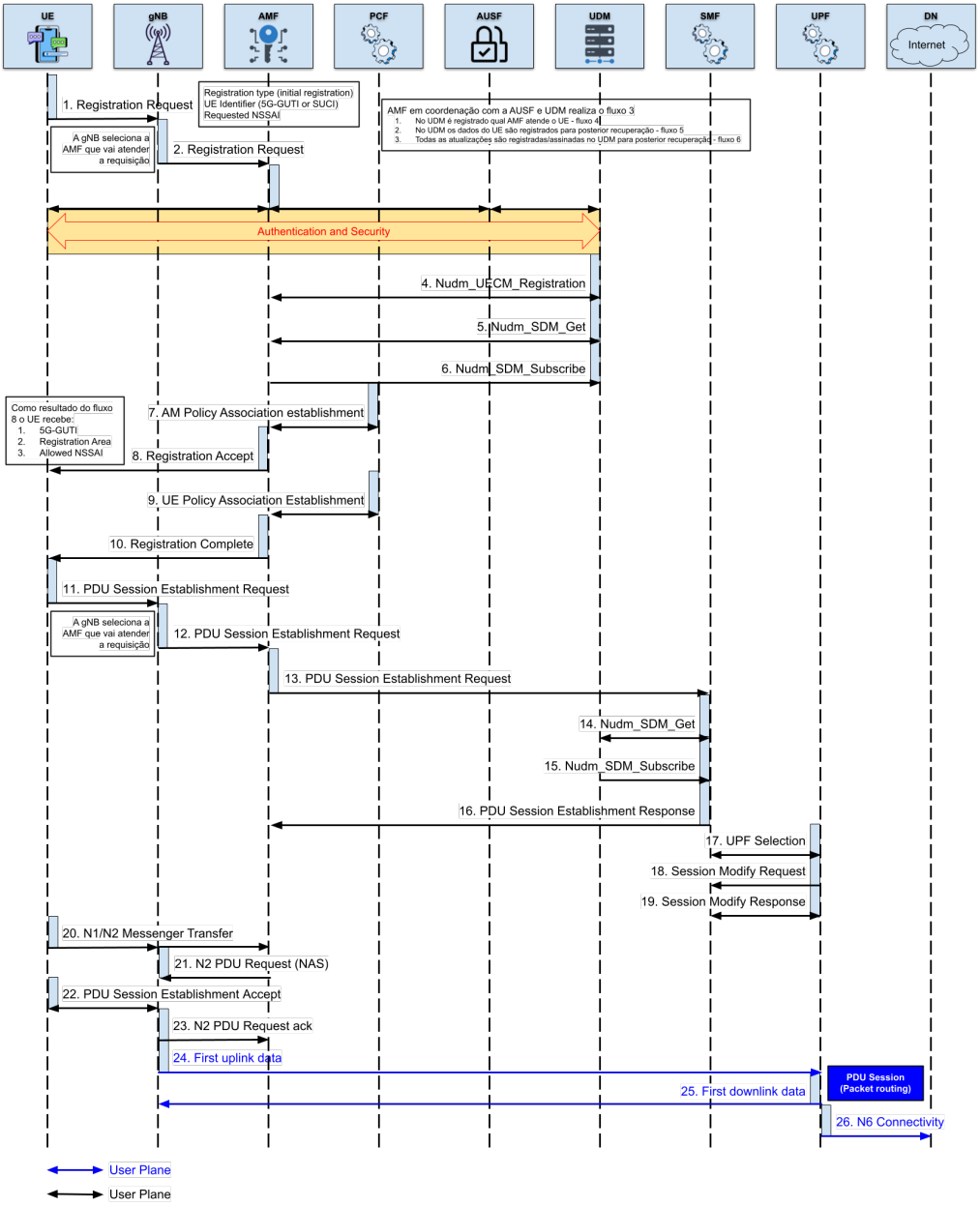

Figura 3.8: *Processo de autenticação de UE e criação e estabelecimento de uma sessão de dados*

As mesmas combinações de compressão e protocolos de transporte apresentados na Tabela 3.1, também foram utilizadas neste experimento. A Tabela 3.3 resume os resultados, os quais também são apresentados nas Figuras 3.9 e 3.10.

Quanto à latência, a Figura 3.9 mostra que o cenário 1 apresenta a maior latência média, de 38 ms, com uma moda de 33 ms. O desvio padrão elevado (18,0721 ms) indica maior variabilidade no tempo de resposta, comprometendo a previsibilidade de resposta da rede para aplicações XR. O cenário 2 registra uma latência média menor, de 34 ms, com a moda também em 33 ms e um desvio padrão reduzido 3,9984 ms. Esses valores indicam maior estabilidade e melhor desempenho quando comparado ao cenário 1. O cenário 3 apresenta um resultado ligeiramente melhor que o cenário 2, com latência média de 33 ms e um desvio padrão de 3,8141 ms. A moda, novamente em 33 ms.

Em relação ao throughput, os resultados apresentados na Figura 3.10 justificam a análise de latência, onde para o cenário 1 apresenta o pior desempenho, com uma média de 17 fps, uma moda de 15 fps e um desvio padrão de 8,2793 fps. Esse cenário sofre com baixa eficiência e alta variabilidade, comprometendo a entrega consistente de quadros de vídeos. O cenário 2 mostra melhorias consideráveis, alcançando uma média de 29 fps, com a moda também em 29 fps e um desvio padrão de 10,6515 fps. Isso demonstra uma entrega mais eficiente de dados, embora ainda com alguma variabilidade. O cenário 3 apresenta o melhor desempenho, com média e moda alinhadas em 30 fps, além de um desvio padrão reduzido de 3,8141 fps. Isso indica uma melhora da eficiência e estabilidade na transmissão.

Cenários dos Experimentos	**End-to-End (E2E) (ms)**			**Throughput (fps)**		
	M	**Mo**	**DP**	**M**	**Mo**	**DP**
1	38,0000	33,0000	18,0721	17,0000	15,0000	8,2793
2	34,0000	33,0000	3,9984	29,0000	29,0000	10,6515
3	33,0000	33,0000	3,8141	30,0000	30,0000	10,0057

Legenda:
M: Média
Mo: Moda
DP: Desvio Padrão

Tabela 3.3: *Resultados do experimento com emuladores*

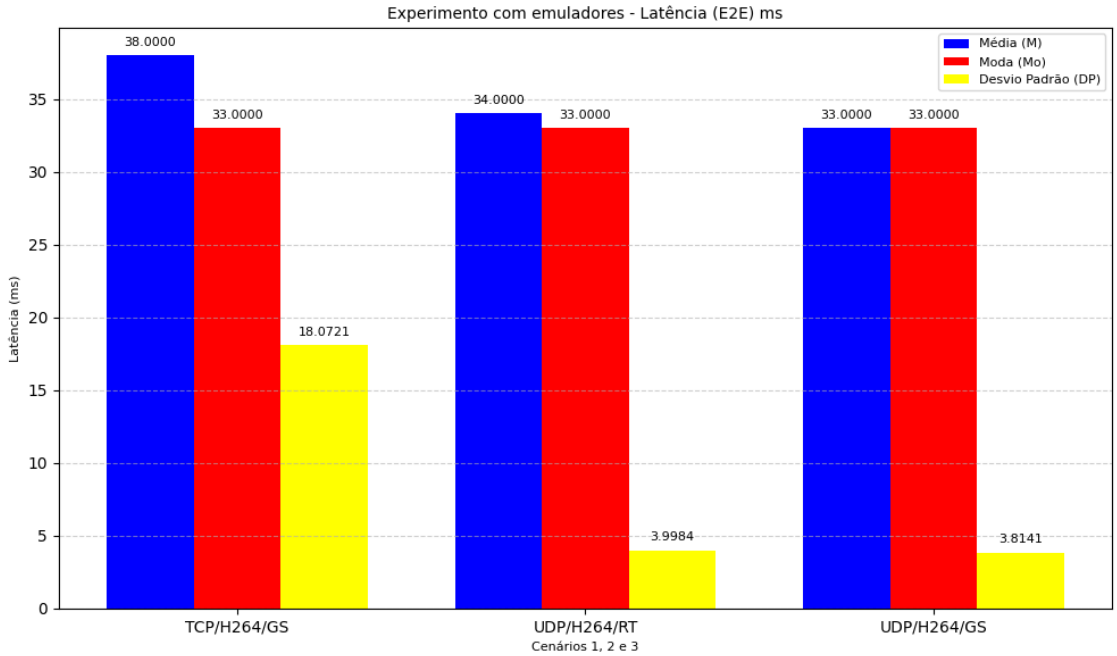

Figura 3.9: ***Latência dos cenários 1, 2 e 3 para o experimento com emuladores***

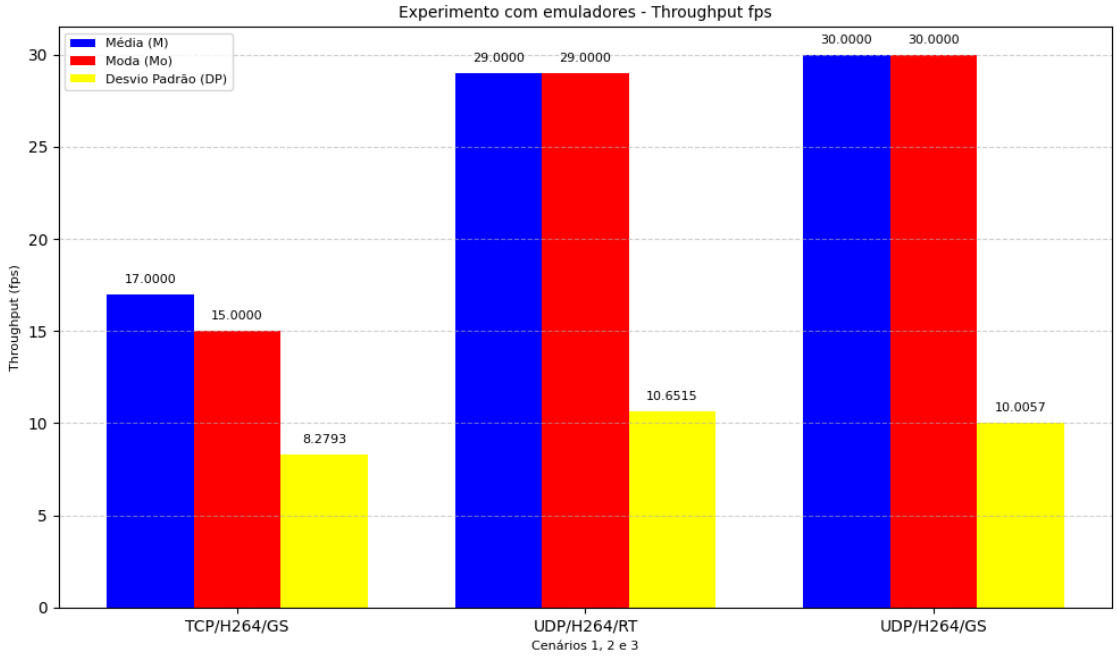

Figura 3.10: ***Throughput dos cenários 1, 2 e 3 para o experimento com emuladores***

No experimento com emuladores, os dados mostram uma relação direta entre latência e throughput. O protocolo UDP, usado nos cenários 2 e 3, se mostrou mais eficiente para aplicações XR, proporcionando menor latência e maior throughput em

comparação ao TCP, que apresentou alta variabilidade e desempenho inferior no cenário 1.

Comparando com os resultados do experimento base podemos perceber comportamentos relativamente semelhantes, onde os resultados destacam que a escolha do protocolo de transporte e do método de compressão tem impacto direto no desempenho da aplicação. Os valores de latência e throughput nos dois experimentos (base e com emuladores) também são semelhantes, demonstrando que, uma vez estabelecida a sessão de dados entre o cliente e o servidor da aplicação, os emuladores não adicionam sobrecarga ao sistema.

3.4.3 Experimento com uma callbox 5G real

Nesta seção, discutimos os resultados obtidos com um *setup* 5G real. De forma semelhante, neste experimento, os mesmos cenários, apresentados na Tabela 3.1, foram utilizados. Utilizamos também parte dos equipamentos do *setup* emulado, substituindo o simulador UERANSIM e o free5GC por uma callbox 5G da Amarisoft, conforme ilustrado na Figura 3.11. A Figura 3.12 detalha os componentes da callbox, a qual consiste em uma solução compacta que reúne todas as funções principais de uma rede móvel, incluindo a rede de acesso a rádio (RAN) e o núcleo da rede, seguindo a especificação 3GPP. Por ser uma solução compacta, esta callbox vem sendo utilizada por empresas, universidades e centros de pesquisa para explorar a tecnologia 5G de maneira simplificada, sem a necessidade de investir em infraestruturas físicas pesadas. Como mostrado na figura, todos os componentes da calbox, com exceção do SDR, consistem em software que se comunicam através de interfaces bem definidas. O componente gNodeB e o SDR implementam a RAN. A gNodeB consiste na parte da RAN que pode ser virtualizada (*virtual RAN*), sendo formada por uma pilha de software, incluindo as camadas L3, L2, L1 e a camada física. Ela se conecta ao SDR, a cabeça de rádio, por meio de uma API aberta. Outro componente da calbox é o núcleo 5G, proprietário da fabricante. O UE (*smartphone*) foi configurado com o chip de acesso para estabelecer a conexão com a RAN. O 5GC autentica o UE e gerencia as sessões, conectando-o à DN para que o cliente MR-Leo acesse o a parte servidora do MR-Leo que executa no servidor de borda.

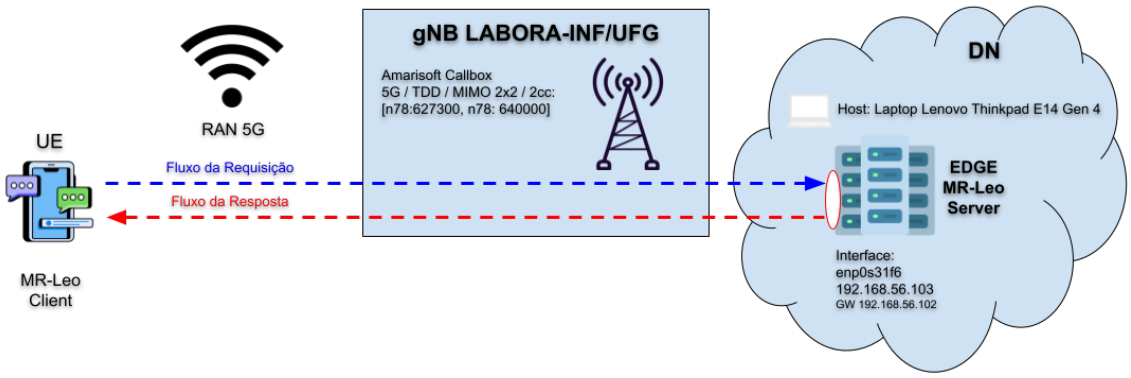

Figura 3.11: *Topologia do experimento com a callbox 5G*

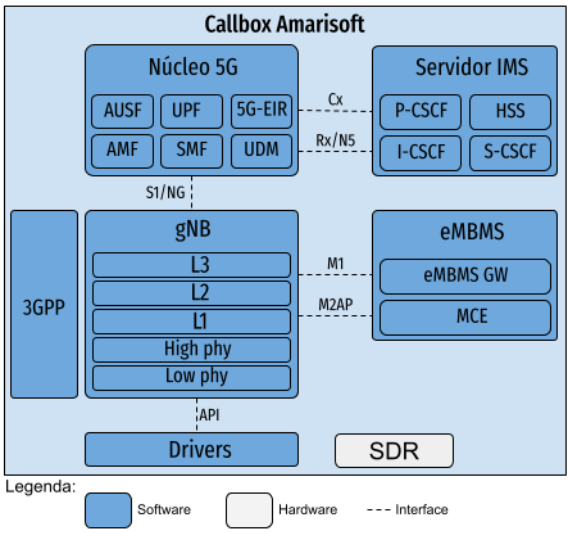

Figura 3.12: *Componentes arquiteturais da callbox 5G*

Cenário dos Experimentos	End-to-End (E2E) (ms)			Throughput (ms)		
	M	Mo	DP	M	Mo	DP
1	FE	FE	FE	FE	FE	FE
2	33,0000	33,0000	34,8309	29,0000	29,0000	8,2370
3	32,0000	32,0000	38,8056	30,0000	30,0000	8,4730

Legenda:
M: Média
Mo: Moda
DP: Desvio Padrão
FE: Falha de Execução

Tabela 3.4: *Resultados do experimento com a callbox 5G*

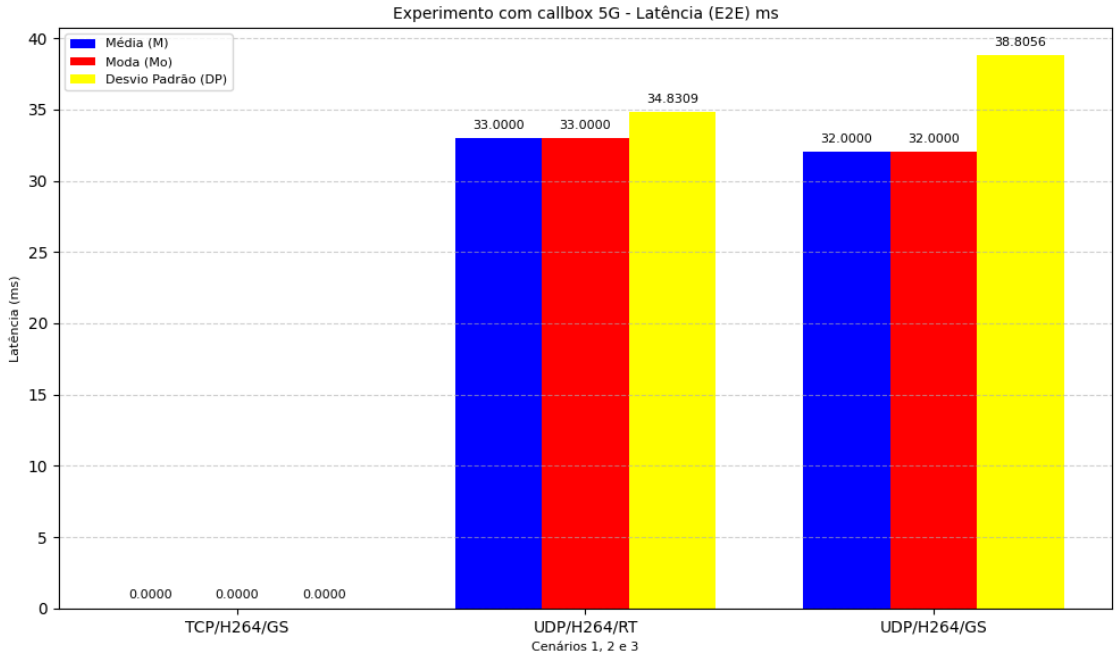

Figura 3.13: ***Latência dos cenários 1, 2 e 3 para o experimento com a callbox 5G***

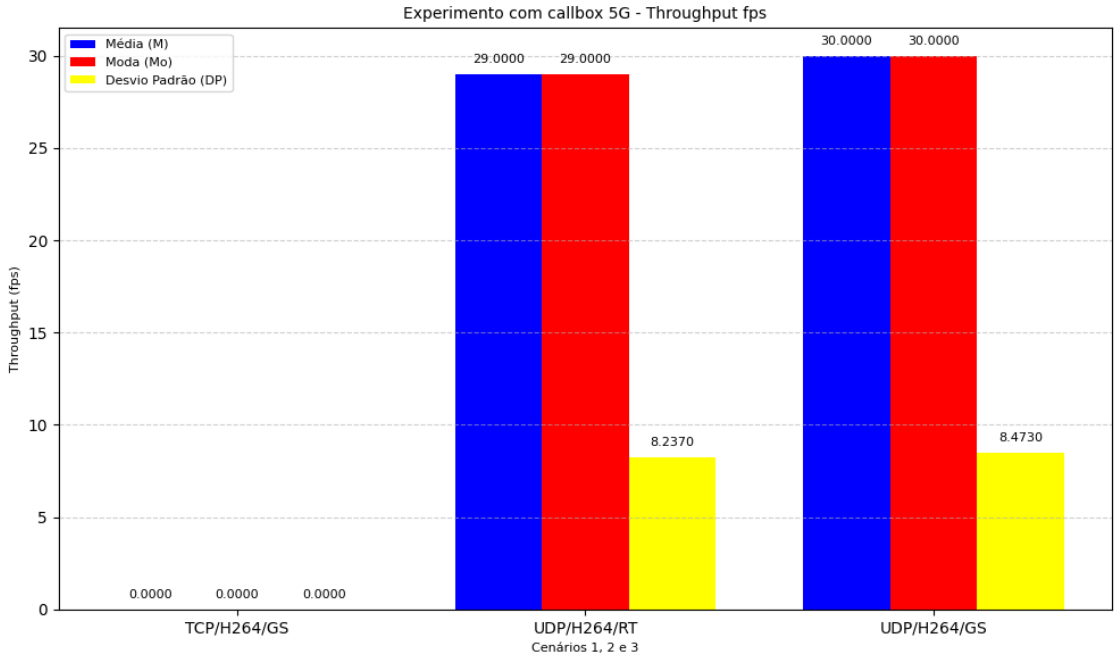

Figura 3.14: ***Throughput dos cenários 1, 2 e 3 para o experimento com a callbox 5G***

A Tabela 3.4 e os gráficos das Figuras 3.13 e 3.14 ilustram os resultados obtidos neste experimento.

A latência média foi um fator crítico analisado nos três cenários. O cenário 1 apresentou uma falha de execução devido à incapacidade do servidor de borda processar

a nuvem de pontos e objetos virtuais em tempo hábil, o que impossibilitou a coleta de dados. Isso demonstra problemas de configuração na callbox ao trabalhar com o protocolo TCP. De fato, a callbox possui centenas de parâmetros e não conseguimos resolver esse problema acionando o suporte da fornecedora em tempo hábil para a conclusão desta dissertação.

Os cenários baseados em UDP não apresentaram falhas. O cenário 2 apresentou uma latência média de 33 ms, com a moda também em 33 ms, indicando consistência nos tempos de resposta. O desvio padrão foi de 34,8309 ms, sugerindo grande variabilidade. O cenário 3 teve a menor latência média, de 32 ms, com a moda igualmente em 32 ms, destacando-se como o cenário mais eficiente em termos de latência. No entanto, o desvio padrão de 38,8056 ms revelou maior variabilidade em relação ao cenário 2. Os gráficos da figura 3.13 corroboram esses resultados, mostrando que os cenários com UDP atingem não falharam, enquanto o TCP não conseguiu lidar com as demandas de processamento.

O throughput, ilustrado na Figura 3.14, reforça as tendências observadas na latência. No cenário 1, o throughput médio não pôde ser medido devido à falha de execução anteriormente citada. O cenário 2 alcançou uma média de 29 fps, com a moda também em 29 fps, indicando consistência na entrega de quadros. O desvio padrão foi de 8,2370 fps, demonstrando variações moeradas no desempenho. O cenário 3 obteve o melhor throughput, com média e moda de 30 fps, além de um desvio padrão de 8,4730 fps, reforçando sua superioridade em termos de entrega eficiente de frames.

É importante observar que os resultados obtidos com a callbox 5G neste trabalho são muito parecidos, em termos de latência e vazão, com os resultados obtidos com o WiFi. Conjecturamos que a causa desse resultado ainda está relacionado com as configurações utilizadas no equipamento. No entanto, uma investigação mais detalhada é necessária para confirmar esta suposição.

CAPÍTULO 4

Conclusões e Trabalhos Futuros

Este capítulo apresenta as considerações finais sobre o trabalho, as produções acadêmicas e reflexões finais sobre trabalhos futuros e direções desta pesquisa.

4.1 Conclusões

As redes 5G, aliadas à computação de borda, representam um avanço na transformação digital, estabelecendo uma base tecnológica para aplicações imersivas e interativas. Essa dissertação explorou de forma abrangente a integração entre essas tecnologias, evidenciando suas potencialidades e os desafios que surgem no contexto de implementação prática. Por meio do modelo 5GMS e de uma aplicação XR, foi possível validar a viabilidade técnica da entrega de serviços. Os resultados qualitativos e quantitativos, reforçam que, ao aproximar os recursos computacionais dos usuários, a computação de borda desempenha um papel importante na superação de limitações de hardware e latência, proporcionando experiências mais fluídas para o usuário final. Entretanto, a complexidade inerente aos arcabouços CAPIF e SEAL, somada às altas demandas de processamento e ao custo de infraestrutura, ainda se apresentaram como desafios significativos para a adoção dessas soluções. Neste contexto, as contribuições apresentadas este trabalho ampliam o entendimento sobre as interações da aplicação e o 5GC.

Apresentamos os fundamentos teóricos e tecnológicos que sustentam essa pesquisa, analisamos trabalhos que apresentaram contribuições para a arquitetura 5G, mas não contemplaram a computação de borda e as aplicações XR. Outros que contribuíram para as aplicações XR usando computação de borda. Mas, uma das contribuições deste trabalho foi abordar os três elementos em conjunto, mostrando os avanços e desafios da utilização da computação de borda na integração de uma aplicação XR nas redes 5G. A análise da arquitetura 5GMS destacou a capacidade de suportar demandas heterogêneas, por meio da SBA. Além disso, a computação de borda foi apresentada como uma solução para reduzir a latência e melhorar o desempenho de aplicações sensíveis ao tempo. O trabalho mostrou o potencial transformador em diferentes experimentos, bem como os desafios relacionados às demandas intensivas de processamento. A integração entre 5G,

computação de borda e aplicações XR demonstrou ser um elemento central para o avanço de aplicações imersivas, permitindo a possibilidade de novos casos de uso. A análise dos trabalhos relacionados revelou uma lacuna importante, tendo em vista que a arquitetura MEC foi definida pela ETSI e os arcabouços CAPIF e SEAL pelo 3GPP, existem sobreposições de funções entre estes elementos o que dificulta uma visão clara do mapeamento para o uso das APIs fornecidas.

A implementação e avaliação qualitativa e quantitativa prática de serviços de MAR, mostrou a viabilidade e o potencial da integração. O modelo 5GMS, demonstrou-se eficiente para suportar os serviços, ao oferecer funcionalidades dedicadas, como o controle de sessões, streaming de mídia e interação em tempo real. Também identificamos pontos convergentes no 5GMS o Aware Application e do Manager no MR-Leo, contudo, diferentemente do modelo 5GMS, o MR-Leo não possui um componente funcionalmente equivalente ao 5GMS AF, o que inviabilizou uma forma nativa de interação com o 5GC. O protótipo MR-Leo se mostrou eficiente ao executar tarefas intensivas na borda da rede, reduzindo a sobrecarga dos dispositivos móveis, garantindo uma melhor experiências para os usuários. Os resultados experimentais, obtidos em cenários emulados e reais, revelaram ganhos significativos em métricas como latência e throughput, evidenciando que a integração entre 5G e computação de borda é uma estratégia possível para superar desafios de limitações técnicas. Além disso, a comparação entre o MR-Leo e o modelo 5GMS confirmou a compatibilidade e a complementaridade entre as propostas, apontando caminhos factíveis para a adoção em larga escala. Este trabalho reforça a importância de abordagens práticas e experimentais para continuar a validar teorias, destacando o impacto do 5G na disponibilidade de aplicações imersivas e na criação de novos paradigmas. Dessa forma, este trabalho contribui também para a evolução da entrega de serviços MAR.

A realização do experimento inicial fundamental para estabelecer uma base de comparação que validasse o desempenho do protótipo. O experimento destacou a importância da escolha de protocolos de transporte e técnicas de compressão de vídeo para atender às demandas de latência e throughput de aplicações MAR. Os resultados mostraram a superioridade do protocolo UDP em relação ao TCP, principalmente na redução da latência e na consistência do fluxo de dados, aspectos críticos para experiências interativas e imersivas em tempo real. A configuração com compressão baseada no GStreamer e transporte por UDP demonstrou ser a mais eficiente, apresentando latência média de 33 ms e throughput estável de 30 fps, atendendo aos KPIs apontados na literatura. Os resultados obtidos serviram como um referencial para os experimentos subsequentes validando a abordagem experimental.

Do experimento com emuladores, usando o Free5GC e do UERANSIM, foi possível replicar cenários complexos e avaliar o desempenho do MR-Leo sob condições con-

troladas. Os resultados evidenciaram a superioridade do protocolo UDP, em particular com a compressão GStreamer, ao garantir baixa latência média e throughput elevado, características essenciais para aplicações XR. O cenário 3 destacou-se como o mais eficiente, apresentando uma latência média de 33 ms e throughput de 30 fps, demonstrando maior estabilidade em comparação aos demais cenários. Esses achados validam a configuração proposta, e reforçam a importância de simulações detalhadas como etapa intermediária na validação de tecnologias, ampliando as possibilidades de otimização e adaptação de aplicações XR em diferentes contextos de uso.

Já no experimento realizado com uma callbox 5G real, confirmou a aplicabilidade prática em um ambiente físico real. Inicialmente os resultados evidenciaram a inadequação do protocolo TCP para aplicações XR, com altas taxas de falha de execução devido às exigências de latência e throughput. Em contraste, os cenários baseados em UDP demonstraram superioridade, particularmente o cenário 3, que alcançou uma latência média de 32 ms e throughput de 30 fps, demonstrando ser a configuração mais eficiente para garantir experiências imersivas e responsivas. Este experimento também revelou os desafios adicionais presentes em redes reais, como interferências ambientais e limitações de hardware, notados por variações no desvio padrão das métricas coletadas. A validação em um ambiente físico reforça a relevância dos testes anteriores em ambientes emulados, consolidando a eficácia do UDP aliado à compressão GStreamer como a configuração ideal para aplicações XR em redes 5G.

4.2 Trabalhos Futuros

Os avanços apresentados nesta dissertação abrem caminhos para diversas oportunidades de pesquisa futura. Pretendemos explorar a implementação de um componente totalmente funcional no protótipo MR-Leo que seja equivalente ao 5GMS Application Function (AF), permitindo uma interação nativa e dinamicamente mais eficiente com o 5GC. Essa modificação tem o potencial de resolver limitações atualmente identificadas durante os testes, como a interação com APIs do 5GC. A inclusão desse componente também possibilitaria uma comparação direta e mais detalhada com o modelo 5GMS, contribuindo para a padronização de soluções.

Outro ponto relevante que pretendemos investigar é a interoperabilidade entre arcabouços como CAPIF, SEAL e MEC. Em estudos futuros pretendemos propor um modelo unificado que elimine sobreposições funcionais e simplifique o mapeamento de APIs, facilitando a adoção dessas tecnologias. Além disso, é essencial investigar formas de reduzir a complexidade de implementação prática desses arcabouços, especialmente em ambientes de baixa capacidade computacional ou com recursos limitados.

Por fim, pretendemos investigar o motivo pelo qual a callbox 5G apresentou desempenho muito próximo dos experimentos emulados.

Referências

[3GPP 2017]3GPP. *3GPP TS 23.501 V15.0.0*. 2017. [Online]. Disponível em: https://www.3gpp.org/ftp/Specs/archive/23_series/23.501. Acessado em: 20 de agosto de 2024.

[3GPP 2019]3GPP. *3GPP Release 15*. 2019. [Online]. Disponível em: https://www.3gpp.org/specifications-technologies/releases/release-15. Acessado em: 04 de fevereiro de 2023.

[3GPP 2020]3GPP. *3GPP Release 16*. 2020. [Online]. Disponível em: https://www.3gpp.org/specifications-technologies/releases/release-16. Acessado em: 04 de fevereiro de 2023.

[3GPP 2020]3GPP. *3GPP TS 23.434 V16.6.0*. 2020. [Online]. Disponível em: https://www.3gpp.org/ftp/Specs/archive/23_series/23.434. Acessado em: 27 de agosto de 2024.

[3GPP 2020]3GPP. *3GPP TS 29.222 V15.8.0*. 2020. [Online]. Disponível em: https://www.3gpp.org/ftp/Specs/archive/29_series/29.222. Acessado em: 26 de agosto de 2024.

[3GPP 2021-a]3GPP. *Service requirements for the 5G system*. 2021–a. [Online]. Disponível em: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107. Acessado em: 04 de fevereiro de 2023.

[3GPP 2021-b]3GPP. *Typical traffic characteristics of media services on 3GPP networks*. 2021–b. [Online]. Disponível em: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3533. Acessado em: 04 de fevereiro de 2023.

[3GPP 2021-c]3GPP. *System architecture for the 5G System (5GS)*. 2021–c. [Online]. Disponível em: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144. Acessado em: 04 de fevereiro de 2023.

[3GPP 2021-d]3GPP. *Study on XR (Extended Reality) Evaluations for NR*. 2021–d. [Online]. Disponível em: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3736. Acessado em: 04 de fevereiro de 2023.

[3GPP 2022-a]3GPP. *3GPP Release 17*. 2022–a. [Online]. Disponível em: https://www.3gpp.org/specifications-technologies/releases/release-17. Acessado em: 04 de fevereiro de 2023.

[3GPP 2022-b]3GPP. *Extended Reality (XR) in 5G*. 2022–b. [Online]. Disponível em: https://www.3gpp.org/ftp/Specs/archive/26_series/26.928/. Acessado em: 04 de fevereiro de 2023.

[3GPP 2023]3GPP. *3GPP TS 26.501*. 2023. [Online]. Disponível em: https://www.3gpp.org/ftp/Specs/archive/26_series/26.501. Acessado em: 20 de agosto de 2024.

[3GPP 2023]3GPP. *Extended Reality (XR) in 5G*. 2023. [Online]. Disponível em: https://www.3gpp.org/ftp/Specs/archive/26_series/26.928. Acessado em: 28 de agosto de 2024.

[Apple 2024]Apple. *Apple ARKit*. 2024. [Online]. Disponível em: https://developer.apple.com/documentation/arkit/. Acessado em: 10 de setembro de 2024.

[Both et al. 2020]BOTH, C. B. et al. Soft5g+: explorando a softwarização nas redes 5g. *Sociedade Brasileira de Computação*, 2020.

[Cao et al. 2023]CAO, J. et al. Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence. *ACM Comput. Surv.*, v. 55, n. 9, 2023.

[Charismiadis et al. 2023]CHARISMIADIS, A.-S. et al. The 3gpp common api framework: Open-source release and application use cases. In: *2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. [S.l.: s.n.], 2023. p. 472–477.

[Contreras et al. 2020]CONTRERAS, L. M. et al. Computing at the edge: But, what edge? In: *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. [S.l.: s.n.], 2020. p. 1–9.

[Cruz, Achir e Viana 2022]CRUZ, P.; ACHIR, N.; VIANA, A. C. On the edge of the deployment: A survey on multi-access edge computing. Association for Computing Machinery, New York, NY, USA, v. 55, n. 5, 2022. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3529758>.

[ETSI 2016]ETSI. *MEC - Framework and Reference Architecture*. 2016. [Online]. Disponível em: https://www.etsi.org/deliver/etsi_gs/mec/001_099/003/01.01.01_60/gs_mec003v010101p.pdf. Acessado em: 20 de agosto de 2024.

[Foukas et al. 2017]FOUKAS, X. et al. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, v. 55, n. 5, p. 94–100, 2017.

[Fragkos et al. 2021]FRAGKOS, D. et al. 5g vertical application enablers implementation challenges and perspectives. In: *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. [S.l.: s.n.], 2021. p. 117–122.

[free5GC 2024]free5GC. *free5GC*. 2024. [Online]. Disponível em: https://free5gc.org/. Acessado em: 20 de agosto de 2024.

[free5GC 2024]free5GC. *GitHub free5GC v3.3.0*. 2024. [Online]. Disponível em: https://github.com/free5gc/free5gc/tree/v3.3.0. Acessado em: 20 de agosto de 2024.

[Gapeyenko et al. 2023]GAPEYENKO, M. et al. *Standardization of Extended Reality (XR) over 5G and 5G-Advanced 3GPP New Radio*. 2023.

[Garcia-Saavedra e Costa-Pérez 2021]GARCIA-SAAVEDRA, A.; COSTA-PéREZ, X. O-ran: Disrupting the virtualized ran ecosystem. *IEEE Communications Standards Magazine*, v. 5, n. 4, p. 96–103, 2021.

[Google 2024]Google. *Google ARCore*. 2024. [Online]. Disponível em: https://developers.google.com/ar?hl=pt-br. Acessado em: 10 de setembro de 2024.

[Gstreamer 2024]Gstreamer. *GStreamer open-source multimedia framework*. 2024. [Online]. Disponível em: https://github.com/GStreamer/gstreamer. Acessado em: 10 de setembro de 2024.

[Gupta e Jha 2015]GUPTA, A.; JHA, R. K. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, v. 3, p. 1206–1232, 2015.

[Hammad et al. 2023]HAMMAD, N. et al. V-light: Leveraging edge computing for the design of mobile augmented reality games. In: *Proceedings of the 18th International Conference on the Foundations of Digital Games*. New York, NY, USA: Association for Computing Machinery, 2023. (FDG '23). ISBN 9781450398558. Disponível em: <https://doi.org/10.1145/3582437.3582456>.

[Huang et al. 2023]HUANG, Z. et al. Standard evolution of 5g-advanced and future mobile network for extended reality and metaverse. *IEEE Internet of Things Magazine*, v. 6, n. 1, p. 20–25, 2023.

[Khalili et al. 2018]KHALILI, H. et al. Design considerations for an energy-aware sdn-based architecture in 5g epon nodes. In: *2018 20th International Conference on Transparent Optical Networks (ICTON)*. [S.l.: s.n.], 2018. p. 1–4.

[Klervie Toczé 2019]Klervie Toczé. *Protótipo de aplicação XR - servidor*. 2019. [Online]. Disponível em: https://gitlab.liu.se/ida-rtslab/public-code/2019_mrleo_video. Acessado em: 16 de setembro de 2024.

[Larsen, Checko e Christiansen 2019]LARSEN, L. M. P.; CHECKO, A.; CHRISTIANSEN, H. L. A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks. *IEEE Communications Surveys Tutorials*, v. 21, n. 1, p. 146–172, 2019.

[Morín, Pérez e Armada 2022]MORíN, D. G.; PéREZ, P.; ARMADA, A. G. Toward the distributed implementation of immersive augmented reality architectures on 5g networks. *IEEE Communications Magazine*, v. 60, n. 2, p. 46–52, 2022.

[Pangolin 2024]Pangolin. *Pangolin is a lightweight portable rapid development library for managing OpenGL display / interaction and abstracting video input*. 2024. [Online]. Disponível em: https://github.com/stevenlovegrove/Pangolin. Acessado em: 10 de setembro de 2024.

[Pereira et al. 2021]PEREIRA, N. et al. Arena: The augmented reality edge networking architecture. In: *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. [S.l.: s.n.], 2021. p. 479–488.

[Raul Mur-Artal, Juan D. Tardos, J. M. M. Montiel and Dorian Galvez-Lopez (DBoW2) 2024] Raul Mur-Artal, Juan D. Tardos, J. M. M. Montiel and Dorian Galvez-Lopez (DBoW2). *Real-Time SLAM for Monocular, Stereo and RGB-D Cameras, with Loop Detection and Relocalization Capabilities*. 2024. [Online]. Disponível em: https://github.com/raulmur/ORB_SLAM2. Acessado em: 10 de setembro de 2024.

[Sanchez et al. 2022]SANCHEZ, A. M. et al. Offering the 3gpp common api framework as microservice to vertical industries. In: *2022 Joint European Conference on Networks and Communications & 6G Summit*. [S.l.: s.n.], 2022. p. 363–368.

[Shah et al. 2020]SHAH, S. P. et al. Service enabler layer for 5g verticals. In: *2020 IEEE 3rd 5G World Forum (5GWF)*. [S.l.: s.n.], 2020. p. 269–274.

[Shi et al. 2016]SHI, W. et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, v. 3, n. 5, p. 637–646, 2016.

[Taleb et al. 2023]TALEB, T. et al. Toward supporting xr services: Architecture and enablers. *IEEE Internet of Things Journal*, v. 10, n. 4, p. 3567–3586, 2023.

[Tangudu et al. 2020]TANGUDU, N. D. et al. Common framework for 5g northbound apis. In: *2020 IEEE 3rd 5G World Forum (5GWF)*. [S.l.: s.n.], 2020. p. 275–280.

[Toczé e et al. 2019]TOCZé, K.; et al. Performance study of mixed reality for edge computing. In: *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*. New York, NY, USA: Association for Computing Machinery, 2019. (UCC'19), p. 285–294. ISBN 9781450368940. Disponível em: <https://doi-org.ez49.periodicos.capes.gov.br/10.1145/3344341.3368816>.

[Toczé e et al. 2020]TOCZé, K.; et al. Characterization and modeling of an edge computing mixed reality workload. *J. Cloud Comput.*, Hindawi Limited, London, GBR, v. 9, n. 1, dec 2020. ISSN 2192-113X. Disponível em: <https://doi-org.ez49.periodicos.capes.gov.br/10.1186/s13677-020-00190-x>.

[Tsolkas e Koumaras 2022]TSOLKAS, D.; KOUMARAS, H. On the development and provisioning of vertical applications in the beyond 5g era. *IEEE Networking Letters*, v. 4, n. 1, p. 43–47, 2022.

[UERANSIM 2024]UERANSIM. *GitHub UERANSIM v3.2.6*. 2024. [Online]. Disponível em: https://github.com/aligungr/UERANSIM/tree/v3.2.6. Acessado em: 20 de agosto de 2024.

[UERANSIM 2024]UERANSIM. *ueransim*. 2024. [Online]. Disponível em: https://github.com/aligungr/UERANSIM. Acessado em: 20 de agosto de 2024.