Brasil 6G

Projeto Brasil 6G
Fase III

Atividade 3.3 - Aplicações de IoT para Redes 6G - Parte 1

15/07/2025

HISTÓRICO DE ATUALIZAÇÕES:

Versão	Data	Autor(es)	Notas
1	15/07/2025	Aldebaro Klautau – UFPA Elton Vivot – UFG Felipe A. P. de Figueiredo – Inatel Felipe Bastos – UFPA Karlla Chaves – UFG Kleber Vieira Cardoso – UFG Pedro M. R. Pereira – Inatel Pedro Sousa – UFPA Rausley A. A. de Souza – Inatel Richard Demo Souza – UFSC Samuel Baraldi Mafra – Inatel Sand Luz Correa – UFG Victoria Dala Pegorara Souto – Inatel	Elaboração de conteúdo
2	15/07/2025	Luciano Leonel Mendes – Inatel Richard Demo Souza – UFSC Vanessa Mendes Rennó – Inatel Victoria Dala Pegorara Souto – Inatel	Revisão de texto

Lista de Figuras

Lista de Tabelas

Acrônimos

3GPP *3rd Generation Partnership Project*

5G Quinta Geração de Rede Móvel Celular

6G Sexta Geração de Rede Móvel Celular

AoA *Angle of Arrival*

AR *Augmented Reality*

ARFL *Angle - Received Signal Strength Indicator Fusion Localization*

AWGN *Additive White Gaussian Noise*

BCE *Binary Cross Entropy*

BEGAN *Boundary Equilibrium Generative Adversarial Network*

BLER *Block Error Rate*

BLE *Bluetooth Low Energy*

BPSK *Binary Phase Shift Keying*

BS *Base Station*

CDF *Cumulative Distribution Function*

CNC *Computing and Network Convergence*

CN *Computing Node*

CNN *Convolutional Neural Network*

CPU *Central Processing Unit*

CSI *Channel State Information*

CUDA *Compute Unified Device Architecture*

DCGAN *Deep Convolutional Generative Adversarial Network*

DL *Deep Learning*

EBF *Energy Beamforming*

EH *Energy Harvesting*

ELU *Exponential Linear Unit*

EMF *Electromagnetic Field*

eMBB *enhanced Mobile Broadband*

eMR-Leo *enhanced Mixed Reality Linköping Edge Offloading*

ER *Energy Receiver*

ESRGAN *Enhanced Super-Resolution Generative Adversarial Network*

ET *Energy Transmitter*

FCC *Federal Communications Commission*

FDD *Frequency Division Duplex*

FoV *Field of View*

FPS *Frames Per Second*

FRTT *Frame Round Trip Time*

GAN *Generative Adversarial Network*

GNSS *Global Navigation Satellite System*

GPU *Graphics Processing Unit*

HMD *Head-Mounted Display*

ICNIRP *International Commission on Non-Ionizing Radiation Protection*

IoT *Internet of Things*

KF *Kalman Filter*

LOS *Line of Sight*

LPIPS *Learned Perceptual Image Patch Similarity*

mAP *Mean Avarage Precision*

MAR *Mobile Augmented Reality*

MARL *Multi-Agent Reinforcement Learning*

MINLP *Mixed-Integer Nonlinear Programming*

MIMO *Multiple-Input Multiple-Output*

MQAM *M-Quadrature Amplitude Modulation*

MR-Leo *Mixed Reality Linköping Edge Offloading*

MRT *Maximum Ratio Transmission*

mMTC *Massive Machine Type Communication*

NLOS *Non-Line of Sight*

NS *Network Slicing*

NVCC *NVIDIA CUDA Compiler*

nvidia-smi *NVIDIA System Management Interface Program*

OFDMA *Orthogonal Frequency Division Multiple Access*

OFDM *Orthogonal Frequency Division Multiplexing*

PDF *Probability Density Function*

PSNR *Peak Signal-to-Noise Ratio*

PTE *Power Transfer Efficiency*

QoE *Quality of Experience*

QoS *Quality of Service*

QPSK *Quadrature Phase Shift Keying*

QuaDRiGa *Quasi Deterministic Radio Channel Generator*

RA *Resource Allocation*

RAN *Radio Access Network*

RAM *Random Access Memory*

RB *Resource Block*

RBG *Resource Block Group*

RCA *Rendering Capacity Allocator*

R-CNN *Region-based Convolutional Neural Network*

RF *Radio Frequency*

RFWPT *Radio Frequency - Wireless Power Transfer*

RIS *Reconfigurable Intelligent Surfaces*

RL *Reinforcement Learning*

RMS *Root Mean Square*

RoI *Region of Interest*

RRS *Radio Resource Scheduling*

RSSI *Received Signal Strength Indicator*

SAC *Soft Actor-Critic*

SC *Semantic Communication*

SGD *Stochastic Gradient Descent*

SINR *Signal-to-Interference-plus-Noise Ratio*

SLA *Service Level Agreements*

SLAM *Simultaneous Localization and Mapping*

SNR *Signal-to-Noise Ratio*

SRGAN *Super-Resolution Generative Adversarial Network*

SSR *Satisfaction Rate*

SSIM *Structural Similarity Index Measure*

T2TF *Track-to-Track Fusion*

TCP *Transmission Control Protocol*

TDoA *Time Difference of Arrival*

TFP *Track Fusion Model with Fused Prediction*

ToA *Time of Arrival*

TTI *Transmission Time Interval*

UDP *User Datagram Protocol*

UE *User Equipment*

UIT União Internacional de Telecomunicações

URLLC *Ultra Reliable Low Latency Communications*

VRAM *Video Random Access Memory*

VR-CG *Virtual Reality Cloud Gaming*

VR-GX *Virtual Reality cloud Gaming Resource Allocation Based on Quality of Experience*

WebRTC *Web Real-Time Communication*

WPT *Wireless Power Transfer*

YOLO *Yolo Only Look Once*

Sumário

1 Introdução

Dando continuidade às atividades desenvolvidas nas Fases I e II do projeto Brasil 6G, a Atividade 3.3 da Fase III concentra-se na proposição e avaliação de soluções tecnológicas voltadas a aplicações de *Internet of Things* (IoT), as quais desempenham papel central no ecossistema da Sexta Geração de Rede Móvel Celular (6G). A crescente demanda por conectividade massiva, com requisitos heterogêneos de latência, confiabilidade, mobilidade e eficiência energética, reforça a importância de arquiteturas e técnicas adaptativas, inteligentes e sensíveis ao contexto. Nesse cenário, a IoT emerge como tecnologia-chave para viabilizar aplicações como cidades inteligentes, saúde conectada, automação industrial e agricultura de precisão. Assim, os avanços reportados nesta fase contribuem diretamente para o desenvolvimento de soluções alinhadas às exigências das redes 6G. Este relatório apresenta os principais resultados das pesquisas realizadas na Atividade 3.3 da Fase III do projeto Brasil 6G até o momento.

O Capítulo 2 apresenta os resultados experimentais da implementação do protótipo *enhanced Mixed Reality Linköping Edge Offloading* (eMR-Leo), desenvolvido para aplicações de *Resource Allocation* (RA) com *offloading* de tarefas na borda. O objetivo é avaliar o impacto de diferentes infraestruturas computacionais e protocolos de transporte no desempenho de tarefas de *Simultaneous Localization and Mapping* (SLAM) e detecção de objetos, essenciais para experiências interativas e responsivas. Os experimentos foram realizados em dois cenários computacionais distintos, com análise de métricas como latência fim a fim, vazão, uso de recursos e consumo energético.

O Capítulo 3 aborda a transmissão de vídeo em redes sem fio sob alta demanda, com foco em aplicações como videoconferência, *cloud gaming* e realidade virtual. Propõe-se um sistema modular baseado em comunicação semântica, que utiliza modelos de aprendizado profundo para extrair e transmitir apenas as informações mais relevantes dos quadros de vídeo, reduzindo o volume de dados e mantendo a qualidade visual. A solução é integrada ao protocolo *Web Real-Time Communication* (WebRTC), permitindo a avaliação de desempenho em condições adversas de rede, bem como sua aplicação em cenários móveis e interativos.

O Capítulo 4 propõe uma abordagem para alocação de recursos em jogos em nuvem com realidade virtual e gráficos imersivos, em conformidade com especificações do *3rd Generation Partnership Project* (3GPP). A solução envolve a adaptação dinâmica da resolução de renderização, taxa de quadros e posicionamento das aplicações em nós distribuídos, buscando maximizar a *Quality of Experience* (QoE) sob restrições de capacidade e latência. O foco está em aplicações *Virtual Reality Cloud Gaming* (VR-CG), com usuários interagindo em tempo real em ambientes virtuais renderizados remotamente, utilizando redes 6G.

O Capítulo 5 apresenta uma arquitetura ciberfísica para sistemas *Radio Frequency - Wireless Power Transfer* (RFWPT), com foco em ambientes internos como escritórios e instalações industriais. A proposta integra otimização da eficiência energética, controle de exposição a campos eletromagnéticos e uso de tecnologias como *Multiple-Input Multiple-Output* (MIMO), *Reconfigurable Intelligent Surfaces* (RIS) e sensores ambientais. O sistema combina *hardware*, protocolos de controle e inteligência artificial para possibilitar carregamento sem fio seguro e eficiente em cenários com elevada demanda por conectividade sustentável.

O Capítulo 6 descreve um sistema de posicionamento *indoor* baseado em *Bluetooth Low Energy* (BLE), que utiliza medições de *Angle of Arrival* (AoA) e *Received Signal Strength Indicator* (RSSI) integradas por meio de filtros de Kalman. A solução é direcionada a ambientes nos quais os sinais de *Global Navigation Satellite System* (GNSS) são ineficazes, com aplicações nos setores de varejo, saúde, manufatura e logística. O sistema busca maior precisão e robustez em

cenários internos complexos, contribuindo para melhorias operacionais, segurança e experiência do usuário.

O Capítulo 7 analisa o desempenho de *autoencoders* baseados em *Convolutional Neural Network* (CNN) para sistemas de comunicação em redes IoT, considerando canais com desvanecimento $\kappa-\mu$. A proposta aplica aprendizado profundo à otimização conjunta de transmissor e receptor, visando maior robustez em ambientes com interferências complexas. Os resultados demonstram ganhos de desempenho em termos de *Block Error Rate* (BLER), especialmente quando comparados a esquemas convencionais, evidenciando o potencial dos *autoencoders* em cenários IoT com restrições severas de recursos.

O Capítulo 8 propõe uma solução baseada em aprendizado profundo para detecção e monitoramento em tempo real de incêndios florestais. A arquitetura utiliza modelos de visão computacional, como YOLOv11, YOLOv8 e *Region-based Convolutional Neural Network* (R-CNN), aliados a sensores IoT, para identificar e acompanhar focos de incêndio em diversos estágios e condições ambientais. A abordagem visa fornecer suporte a decisões rápidas e eficazes, sendo especialmente relevante em regiões de difícil acesso e alta vulnerabilidade ambiental.

O Capítulo 9 apresenta uma solução para orquestração de recursos de rádio em redes 6G voltadas à Indústria 4.0, com uso de *Multi-Agent Reinforcement Learning* (MARL). A arquitetura hierárquica proposta, baseada no algoritmo *Soft Actor-Critic* (SAC), gerencia a alocação dinâmica de recursos entre *slices* e dentro de cada *slice*, com ênfase em serviços críticos como *Ultra Reliable Low Latency Communications* (URLLC). São detalhados a modelagem do ambiente, o treinamento dos agentes e os cenários de simulação, evidenciando melhorias em vazão, latência e taxa de entrega de pacotes.

Por fim, o Capítulo 10 apresenta as conclusões gerais do relatório.

2 Modelo de Carga de Trabalho para Aplicações de Realidade Aumentada Móvel

2.1 Introdução

Na última década, avanços significativos impulsionaram a evolução da *Augmented Reality* (AR), como o lançamento do *Google Glass* em 2013 e do *HoloLens* em 2016, consolidando a tecnologia como uma solução promissora e em constante desenvolvimento. Uma tendência emergente neste contexto é o uso da AR em dispositivos móveis, ou *User Equipment* (UE), como *smartphones* e *Head-Mounted Display* (HMD). Essas soluções são conhecidas como aplicações móveis de realidade aumentada, ou *Mobile Augmented Reality* (MAR) [1], e têm atraído o interesse de diversos setores da economia. Na área de navegação, por exemplo, aplicações MAR podem ser utilizadas em *smartphones* para reconhecer objetos no ambiente e sobrepor instruções diretamente na visão do usuário. Na Indústria 4.0, essas aplicações se destacam como habilitadores-chave, com usos em manutenção remota, colaboração à distância e treinamentos interativos voltados a operadores no chão de fábrica [2]. Na medicina, aplicações MAR também apresentam grande potencial, oferecendo novas abordagens para a relação médico-paciente, tratamentos e educação [3].

A Figura 1 ilustra o fluxo típico de uma aplicação MAR. De modo geral, esse tipo de aplicação é composto por dois componentes principais: as operações de entrada e saída e as tarefas específicas de MAR [1]. As operações de entrada e saída envolvem o uso de sensores como câmeras, acelerômetros, giroscópios, sensores de profundidade, entre outros. Esses sensores funcionam como uma ponte entre o mundo físico e o ambiente virtual, capturando informações do ambiente físico em tempo real (Passo 1). Os dados provenientes desses sensores são então encaminhados para processamento pelas tarefas MAR (Passo 2). Dois módulos principais realizam esse processamento: SLAM e detecção de objetos. O módulo de SLAM permite estimar simultaneamente a posição do UE no espaço e construir um mapa tridimensional do ambiente. O módulo de detecção de objetos, por sua vez, é responsável por identificar e classificar os elementos visuais presentes em um quadro ou cena. Esses dois módulos podem operar de forma independente ou integrada, dependendo da arquitetura da aplicação. As saídas desses módulos são utilizadas para criar objetos virtuais com diferentes formas — como modelos tridimensionais, caixas delimitadoras (*bounding boxes*), nuvens de pontos, entre outros — e são então alinhados espacialmente com base na localização e orientação do usuário (Passo 3). Em seguida, essa informação é sobreposta às imagens do mundo real capturadas (Passo 4). Por fim, o conteúdo aumentado é encaminhado ao dispositivo de saída, permitindo que o usuário visualize a cena com elementos virtuais integrados ao ambiente físico (Passo 5).

Os módulos de SLAM e detecção de objetos utilizam algoritmos complexos de visão computacional, os quais normalmente exigem grande poder de processamento. No entanto, os UEs oferecem capacidades computacionais e/ou de energia limitadas para executar esses módulos com alta qualidade de experiência. Uma forma de resolver esse problema é descarregar (*offload*) essas tarefas computacionalmente intensivas para servidores localizados na borda da rede. Para o usuário final, o descarregamento de tarefas para a borda significa economia de energia e redução de aquecimento no dispositivo. O descarregamento de tarefas críticas para uma infraestrutura de borda pode também aumentar a qualidade da experiência do usuário, fornecendo às tarefas acesso a aceleradores de *hardware* como *Graphics Processing Unit*s (GPUs).

O conteúdo deste capítulo foi desenvolvido pelos pesquisadores *Karlla Bianca Chaves Rodrigues, Kleber Vieira Cardoso e Sand Luz Correa.*

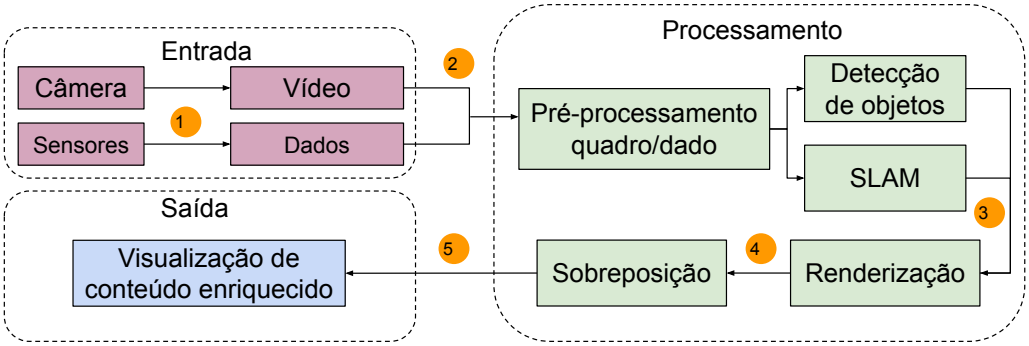


Figura 1: Arquitetura básica para aplicações MAR.

Ao mesmo tempo, o *offloading* pode trazer benefícios aos desenvolvedores de MAR que podem aproveitar um serviço de descarregamento para tratar questões de privacidade e otimizar o desempenho com base nos requisitos do usuário e em fatores contextuais [4].

Para que o descarregamento de tarefas seja eficaz, é fundamental compreender o perfil de consumo de recursos de cada tarefa em diferentes arquiteturas de *hardware*. Tarefas como SLAM e detecção de objetos apresentam comportamentos computacionais distintos dependendo da plataforma em que são executadas, podendo variar significativamente em termos de uso de *Central Processing Unit* (CPU), GPU, memória e largura de banda. Essa variabilidade torna essencial a realização de análises de desempenho que considerem não apenas o tempo de execução e o consumo energético, mas também a compatibilidade com aceleradores específicos, como unidades de processamento gráfico ou motores de inferência neural. Um mapeamento preciso desses perfis permite que os mecanismos de descarregamento tomem decisões informadas sobre quando, onde e como executar determinadas tarefas, maximizando o desempenho e a eficiência energética, ao mesmo tempo em que atendem a restrições contextuais, como latência e conectividade da rede. Assim, o conhecimento detalhado sobre o comportamento das tarefas em diferentes arquiteturas se torna um componente estratégico no desenvolvimento de aplicações MAR adaptativas e eficientes.

Apesar de importante, na literatura, apenas o trabalho em [5] provê um modelo de carga para aplicações MAR extraída de um protótipo real, denominado *Mixed Reality Linköping Edge Offloading* (MR-Leo). Ainda assim, apenas a tarefa de SLAM é representada neste modelo. Adicionalmente, em termos de processamento, apenas a carga de trabalho de CPU é caracterizada. Diante dessa lacuna, este trabalho tem como objetivo avançar o estado da arte na caracterização de cargas de trabalho de aplicações MAR, por meio das seguintes contribuições:

- O protótipo MR-Leo proposto em [5] foi modificado visando adicionar uma tarefa de SLAM acelerada por GPU;
- Foi adicionado ao protótipo uma tarefa de detecção de objetos baseada na biblioteca *Yolo Only Look Once* (YOLO) que pode ser executada tanto em CPU quanto em GPU;
- A partir das implementações desenvolvidas, foi analisado o desempenho das tarefas de SLAM e detecção de objetos em duas infraestruturas de borda distintas: com e sem GPU;
- Para cada tarefa, foi derivado um novo modelo de carga de trabalho levando em consideração a infraestrutura que se mostrou mais adequada para a execução da tarefa.

2.2 Solução Proposta

MR-Leo [5] é um protótipo desenvolvido na Universidade de Linköping, na Suécia, o qual tem como objetivo habilitar aplicações de realidade mista, particularmente AR, por meio da integração com a computação de borda. Conforme ilustrado na Figura 2, a arquitetura do MR-Leo é composta por dois componentes principais: o serviço executado no servidor de borda e o aplicativo em execução no UE. Grande parte do protótipo envolve a transmissão de vídeo. A partir do UE, as imagens capturadas pela câmera são convertidas em um fluxo de vídeo pelo aplicativo e enviadas ao servidor de borda. O serviço de borda recebe os quadros e os armazena em uma fila. Além de lidar com a recepção e transmissão de vídeo, esse serviço executa diversas tarefas, sendo a principal a de realidade mista, composta por duas sub-tarefas: SLAM e sobreposição gráfica. A tarefa de SLAM retira quadros da fila e os processa para gerar um mapa tridimensional do ambiente, representado por uma nuvem de pontos. O SLAM também determina a posição e a rotação do UE no modelo (mapa), conhecidas como pose. Quando uma nuvem de pontos válida é criada e a pose do dispositivo é conhecida, é possível adicionar objetos gráficos tridimensionais ao ambiente. A decisão de inserir esses objetos é feita pelo usuário, por meio de um comando enviado ao servidor de borda. Ao acionar esse comando, a nuvem de pontos e a pose são utilizadas como entrada pela tarefa de sobreposição gráfica, que cria os objetos virtuais e os posiciona corretamente no espaço mapeado, gerando uma sobreposição sobre o quadro original. Por fim, o quadro resultante, composto pela imagem original com os elementos virtuais adicionados, é renderizado e enviado de volta ao UE para exibição ao usuário.

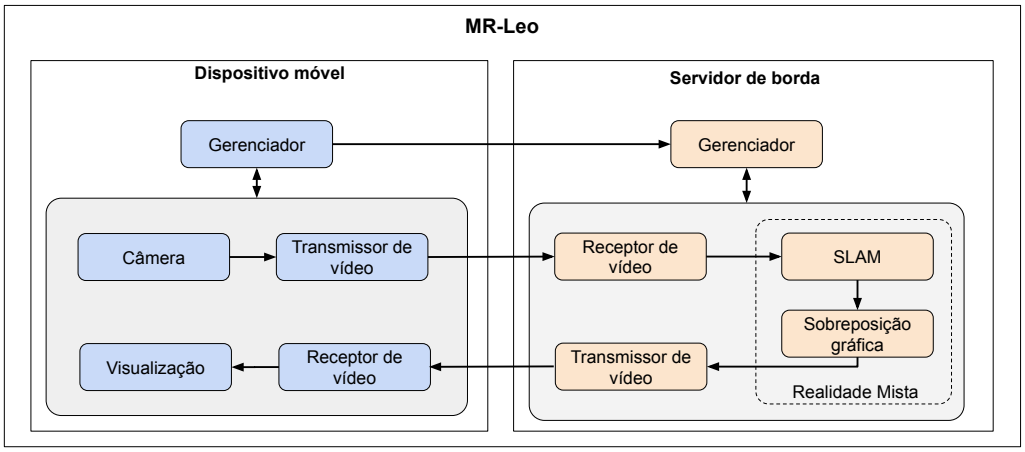


Figura 2: Principais componentes do MR-Leo.

Como descrito acima, o aplicativo que executa no UE não realiza nenhuma análise do conteúdo das imagens capturadas pela câmera; ele apenas as codifica como vídeo e as envia para o servidor de borda. As tarefas de SLAM, renderização e sobreposição gráfica são implementadas pelo serviço de borda. O serviço de borda do MR-Leo é implementado em C++. O protótipo usa o ORB-SLAM2 [6] como biblioteca de SLAM. O serviço de transmissão e recepção de vídeo tanto no cliente como no servidor usa o *Gstreamer* e a renderização é feita pela biblioteca *Pangolin*. O aplicativo cliente é implementado em Java, tendo sido desenvolvido para a plataforma Android. O aplicativo cliente e o serviço de borda se comunicam via *socket Transmission Control Protocol* (TCP) ou *User Datagram Protocol* (UDP), enquanto H.264 é o formato de compressão de vídeo usado em ambas as implementações.

Tendo como base o protótipo do MR-Leo, este trabalho propôs um novo protótipo denominado eMR-Leo. Para desenvolvimento do eMR-Leo foi realizada uma modificação no código do MR-Leo para que o ORB-SLAM2 pudesse executar em GPU. Além disso, o protótipo MR-Leo foi alterado para adicionar a tarefa de detecção de objetos, a qual pode ser executada em CPU e GPU.

Para o desenvolvimento da solução proposta, a biblioteca ORB-SLAM2 originalmente utilizada no MR-Leo foi substituída pela versão *Compute Unified Device Architecture* (CUDA) *accelerated* ORB-SLAM [7]. Essa versão modifica o ORB-SLAM2 para explorar a computação paralela oferecida pela GPU por meio do CUDA, com ênfase na etapa de extração de características. Entre as otimizações implementadas, destaca-se a construção paralela da pirâmide de imagens. Para viabilizar essa substituição, o arquivo de configuração responsável pela definição da construção do projeto MR-Leo também foi ajustado, de modo a incluir as instruções necessárias para acionar o compilador *NVIDIA CUDA Compiler* (NVCC). Com isso, tornou-se possível integrar os objetos CUDA à implementação do MR-Leo, mantendo a consistência do fluxo de desenvolvimento e execução da aplicação.

Em seguida, a tarefa de detecção de objetos foi adicionada ao protótipo original, usando o YOLO, um algoritmo de rede neural profunda usado para detecção de objetos em tempo real. Como o MR-Leo é implementado em C++, foi utilizada a *Darknet* e a *Darkhelp* para integrar o YOLO ao código do protótipo. A primeira consiste numa implementação em C++ e CUDA do YOLO, enquanto a segunda é um *wrapper* que facilita o uso da *Darknet* em programas C++. Essa integração ocorreu estendendo a classe `ImageProcesser`.

2.3 Resultados Experimentais

Nesta seção são apresentados os resultados obtidos pelo protótipo eMR-Leo proposto neste trabalho. Inicialmente, foi realizada uma avaliação do desempenho das duas tarefas — SLAM e detecção de objetos — empregando diferentes infraestruturas de computação na borda. Adicionalmente, analisou-se o impacto dessa utilização sobre a vazão e a latência fim a fim do protótipo.

2.3.1 Cenários de Avaliação e Ambiente de Testes

Para garantir condições equivalentes em diferentes execuções de um mesmo experimento, foi utilizado um vídeo pré-gravado com 60 segundos de duração, em substituição à captura de imagens em tempo real. O vídeo apresenta resolução de 640×480 *pixels* e taxa de 30 *Frames Per Second* (FPS). Cada quadro é submetido ao seguinte fluxo de processamento: (a) reprodução em um *smartphone*, (b) codificação, (c) transmissão para o servidor de borda, (d) decodificação, (e) processamento — com a inserção de elementos virtuais —, (f) renderização, (g) nova codificação, (h) retransmissão ao UE com os elementos virtuais incorporados, (i) decodificação e, por fim, (j) exibição ao usuário. Esse *pipeline* é ilustrado na Figura 3.

No contexto da tarefa de SLAM, o enriquecimento consiste na geração de uma nuvem de pontos e sua sobreposição sobre o quadro processado. Já na tarefa de detecção de objetos, o enriquecimento ocorre por meio da inserção de caixas delimitadoras (*bounding boxes*) ao redor dos objetos identificados. Cada experimento descrito na avaliação a seguir é executado 30 vezes.

---

https://pjreddie.com/darknet/
https://github.com/stephanecharette/DarkHelp

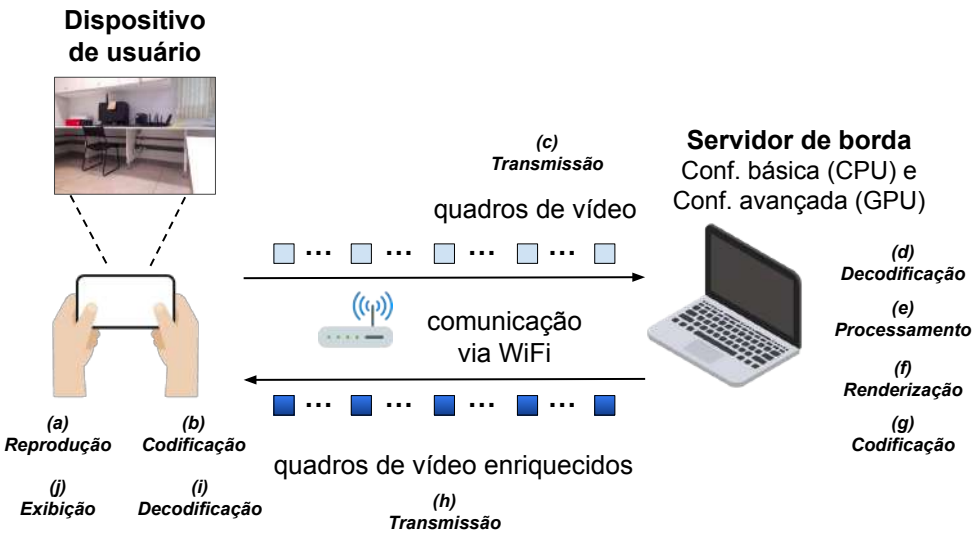


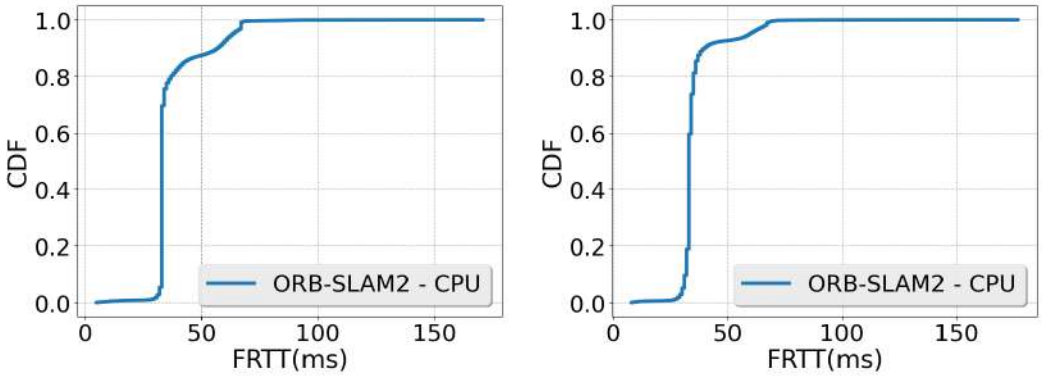
Figura 3: Ambientes de experimentação utilizados.

Para entender como as tarefas de SLAM e detecção de objetos utilizam diferentes recursos computacionais e de comunicação, foram considerados dois ambientes de experimentação. O primeiro, denominado **configuração básica**, utiliza um *laptop* Dell G15, equipado com um processador Intel Core i5-13450HX de 6 núcleos e 24 GB de *Random Access Memory* (RAM) DDR5, como servidor de borda. O UE é representado por um *smartphone* Samsung Galaxy A21s com processador octa-core Exynos 850 e RAM de 4 GB, executando o sistema operacional Android na versão 12. Esse dispositivo se conecta ao servidor de borda usando uma rede Wi-Fi de 2,4GHz (802.11b). O segundo, denominado **configuração avançada**, usa um *laptop* Lenovo Legion Slim 5, com um processador Intel Core i5-13420H de 8 núcleos e 16 GB de RAM DDR5 e uma GPU NVIDIA GeForce RTX 3050, com 6 GB de memória, como servidor de borda. Similar a configuração básica, o UE é o *smartphone* Samsung Galaxy A21s, o qual se conecta ao servidor de borda usando a mesma rede Wi-Fi de 2,4GHz. Em ambas as configurações, o servidor de borda executa o Ubuntu 24.04 LTS como sistema operacional. A Figura 3 ilustra as duas configurações utilizadas neste trabalho.

2.3.2 SLAM

Inicialmente, foi avaliada a latência fim-a-fim, por meio da métrica *Frame Round Trip Time* (FRTT), bem como a vazão da tarefa de SLAM, considerando os protocolos de transporte TCP e UDP. As Figuras 4a e 4b exibem as funções de distribuição cumulativa, ou *Cumulative Distribution Functions* (CDFs), para o FRTT para a configuração básica. No protocolo TCP, o FRTT de 80% das amostras ficou abaixo de 38 ms, enquanto no UDP essa estatística alcançou 42 ms, evidenciando, portanto, que não houve um impacto significativo do protocolo de transporte utilizado na latência fim-a-fim.

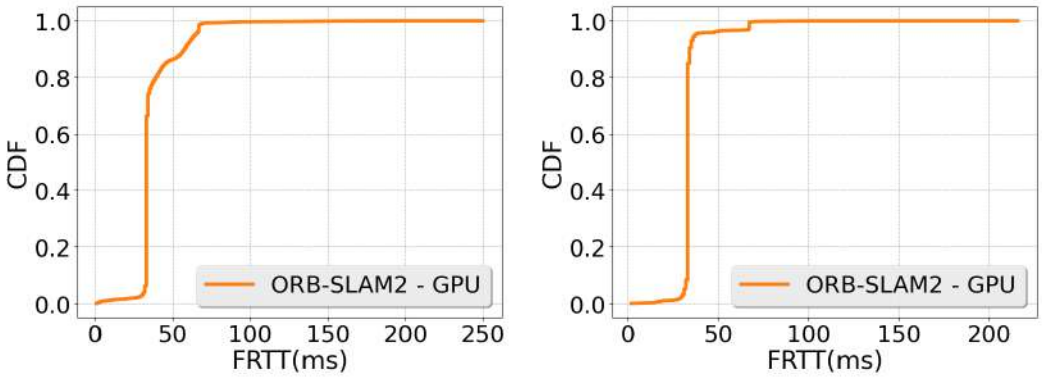
O FRTT referente à tarefa de SLAM na configuração avançada é apresentado na Figura 5. Utilizando o protocolo TCP, verificou-se que 80% das amostras apresentaram FRTT inferior a 38 ms, enquanto com o protocolo UDP esse valor foi de 37 ms. Novamente, observa-se que o protocolo de transporte adotado não exerceu impacto significativo sobre a latência fim-a-fim. De forma análoga, a utilização de GPU na execução da tarefa de SLAM também não resultou em variações significativas na latência fim-a-fim. Esses resultados estão em conformidade com



(a) TCP (b) UDP

Figura 4: Valores de FRTT para a tarefa de SLAM utilizando os protocolos TCP e UDP para a configuração básica.

os dados reportados na literatura [8], que indicam que, mesmo em versões modificadas, como o CUDA-*accelerated* ORB-SLAM, os ganhos de desempenho decorrentes do uso de GPU podem ser inferiores ao esperado ou, em alguns casos, praticamente nulos. Isso ocorre porque, mesmo com a aceleração de etapas específicas — como a extração de características —, grande parte da lógica do ORB-SLAM2 continua sendo executada na CPU. Isso inclui o rastreamento da pose, o ajuste e a otimização do mapa, a construção e manutenção do grafo de mapas, além da detecção de *loops*.



(a) TCP (b) UDP

Figura 5: Valores de FRTT para a tarefa de SLAM utilizando os protocolos TCP e UDP para a configuração avançada.

A Tabela 1 ilustra a vazão média, em termos de FPS obtida na tarefa de SLAM. Novamente, os resultados demonstram que o protocolo e a infraestrutura de computação utilizados, não tiveram impacto significativo na latência fim-a-fim e na vazão da tarefa de SLAM.

Considerando a latência fim-a-fim, o limite aceitável varia conforme o dispositivo utilizado. Por exemplo, para *smartphones*, a latência máxima tolerável é de 100 ms [5], enquanto dispositivos de visualização por HMDs exigem uma latência de, no máximo, 15 ms para evitar enjoos [9]. Assim, os valores de latência fim-a-fim obtidos durante a execução da tarefa de SLAM são considerados adequados. Em relação à vazão, a literatura geralmente adota 24 FPS como o limite mínimo aceitável. Considerando esse critério, os valores de vazão obtidos para a

Tabela 1: Vazão Média da Tarefa de SLAM em FPS.

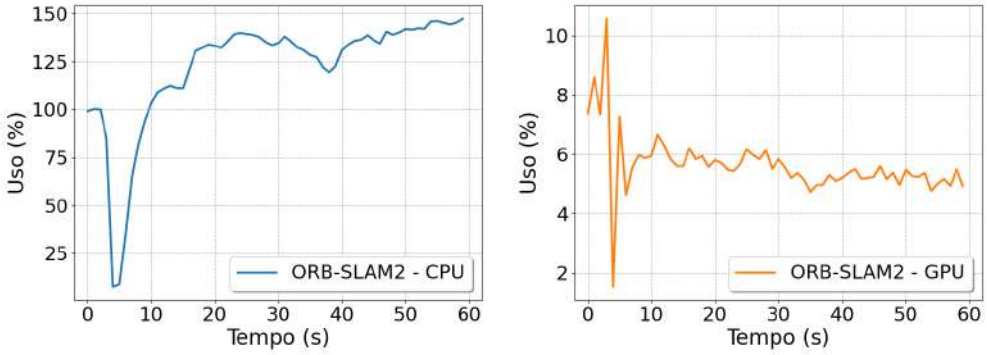
	TCP		UDP	
Configuração	**Média**	**Desvio Padrão**	**Média**	**Desvio Padrão**
Básica	27	1,75	22	8,49
Avançada	27	1,64	29	0,46

tarefa de SLAM mostram-se adequados, exceto na configuração básica com o protocolo UDP, que atingiu apenas 22 FPS.

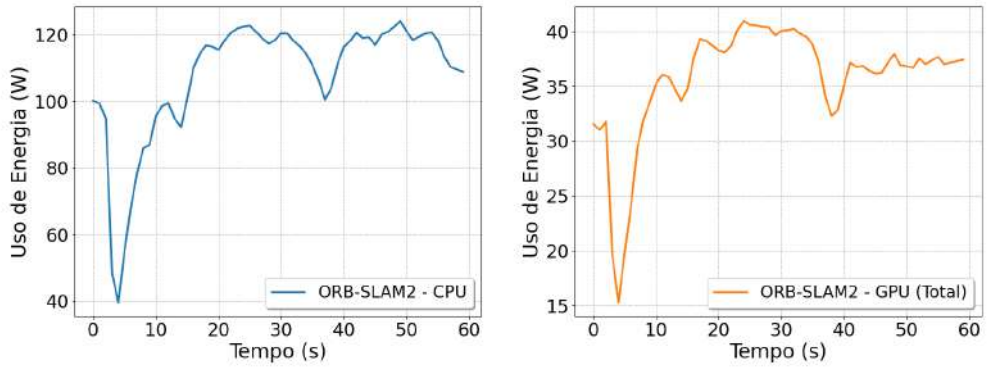
A Figura 6 ilustra o uso de recursos computacionais e o consumo energético do servidor de borda durante a execução da tarefa de SLAM ao longo de toda a transmissão do vídeo. Na Figura 6a, são apresentados os dados referentes à utilização da CPU e da GPU. Na configuração básica, o uso máximo da CPU atingiu 147%, o que indica que a tarefa utilizou 100% de um núcleo e aproximadamente 47% de outro. Isso significa que o processamento exigiu efetivamente dois dos oito núcleos disponíveis no servidor de borda. Já na configuração avançada, o consumo de GPU permaneceu baixo em comparação ao da CPU, alcançando no máximo 10% de utilização. Esses resultados indicam que o uso da GPU pelo ORB-SLAM2 foi modesto, reforçando que a maior parte das operações computacionais ainda é processada pela CPU. Mesmo com a incorporação de acelerações específicas, o algoritmo não delega significativamente o processamento à GPU.

A Figura 6b ilustra o consumo energético do protótipo proposto. Na configuração básica, observou-se um pico de consumo energético de aproximadamente 120 W no momento de maior utilização da CPU. Já na configuração avançada, o consumo máximo registrado foi de cerca de 40 W, coincidindo com o maior uso da GPU. Nessa configuração, o consumo energético foi obtido pela soma das medições coletadas pelas ferramentas *NVIDIA System Management Interface Program* (nvidia-smi) e *powerstat*, sendo a primeira responsável pelo monitoramento do consumo da GPU, e a segunda, da CPU. Assim, o valor apresentado representa o consumo energético total do servidor de borda na configuração avançada, e não apenas o da GPU de forma isolada. A diferença observada entre os consumos energéticos das duas configurações pode ser atribuída aos modelos distintos dos servidores de borda utilizados, sendo o equipamento da configuração básica mais antigo e, consequentemente, menos eficiente do ponto de vista energético.

Por fim, é avaliado o consumo de memória nos servidores de borda ao longo da transmissão do vídeo. Especificamente, a Figura 7a apresenta o consumo de memória RAM na configuração básica, enquanto a Figura 7b ilustra o consumo de memória RAM e *Video Random Access Memory* (VRAM) na configuração avançada. Na configuração básica, observa-se um rápido aumento no consumo de RAM nos primeiros 10 segundos de execução, decorrente do carregamento do vocabulário inicial, bem como das operações de extração de características e estimação de pose realizadas nos quadros iniciais. Após, o ORB-SLAM2 entra em um estado de rastreamento contínuo, onde os dados são processados geralmente um quadro por vez. Neste estado, o número de pontos no mapa espacial pode crescer gradualmente, mas o uso de memória é incremental e muito menor que no início. Na configuração avançada, observa-se o mesmo comportamento no uso de RAM. O uso de VRAM também é maior nos primeiros segundos da transmissão, caindo drasticamente na fase de rastreamento, onde a extração de características (parte executada na GPU) é menos frequente.



(a) Utilização de recursos



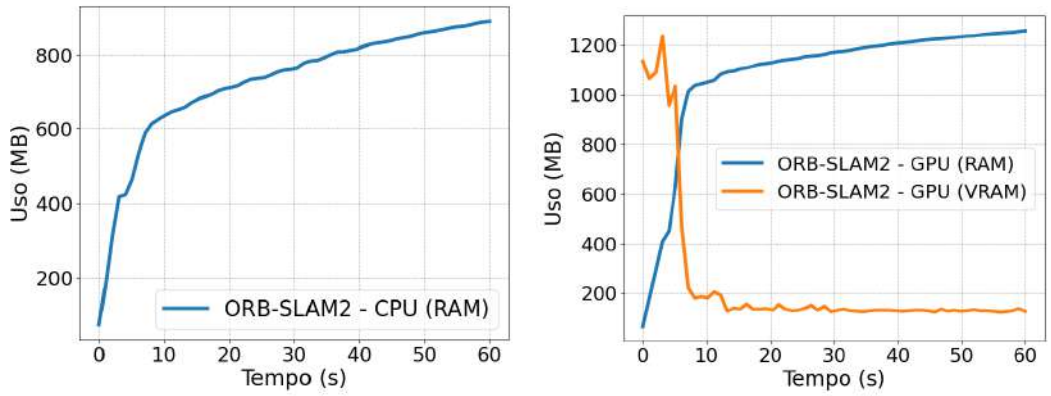
(b) Consumo energético

Figura 6: Consumo de recursos de energia da tarefa de SLAM nas configurações básica (CPU) e avançada (GPU).

2.3.3 Detecção de objetos

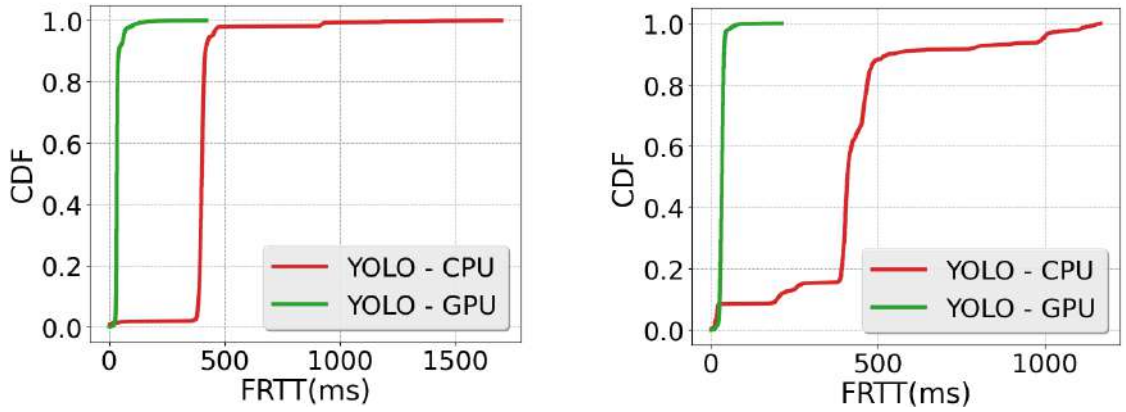
Para a tarefa de detecção de objetos, inicialmente, foi avaliada a latência fim-a-fim, medida através do FRTT. Similar a tarefa de SLAM, considera-se o FRTT usando o protocolo TCP e UDP nas configurações básica e avançada. A Figura 8a exibe as CDFs para o FRTT para a configuração básica (CPU) e avançada (GPU) usando o TCP. Na configuração avançada, 90% das amostras apresentaram FRTT inferior a 41 ms, enquanto na configuração básica esse valor foi inferior a 470 ms. Tal resultado evidencia que, diferentemente do SLAM, a detecção de objetos é fortemente influenciada pela arquitetura de processamento da borda, ou seja, a latência fim-a-fim da tarefa diminui significativamente quando a tarefa é executada em *hardware* com aceleração. Essa diferença decorre do fato de que o YOLO é uma rede convolucional profunda, cujas operações de convolução apresentam alta paralelização. As CDFs do FRTT para as configurações básica e avançada utilizando o protocolo UDP são apresentadas na Figura 8b. Novamente, observa-se significativa redução na latência fim-a-fim da tarefa com o uso da arquitetura equipada com GPU. O protocolo de comunicação empregado, por sua vez, não exerce impacto relevante sobre os resultados, comportamento também observado na tarefa de SLAM.

Também foi realizada a análise da vazão média, em termos de FPS, referente à tarefa de detecção de objetos, conforme apresentado na Tabela 2. Considerando os valores aceitáveis de latência fim-a-fim e taxa de quadros por segundo discutidos anteriormente — 100 ms e 24 FPS, respectivamente — apenas a configuração avançada pode proporcionar uma experiência adequada para o usuário final quando ele faz uso de detecção de objetos. Na configuração



(a) Configuração básica (b) Configuração avançada

Figura 7: Consumos de RAM (configuração básica) e RAM e VRAM (configuração avançada para a tarefa de SLAM.



(a) TCP (b) UDP

Figura 8: Valores de FRTT para a tarefa de detecção de objetos utilizando os protocolos TCP e UDP para as configurações básica e avançada.

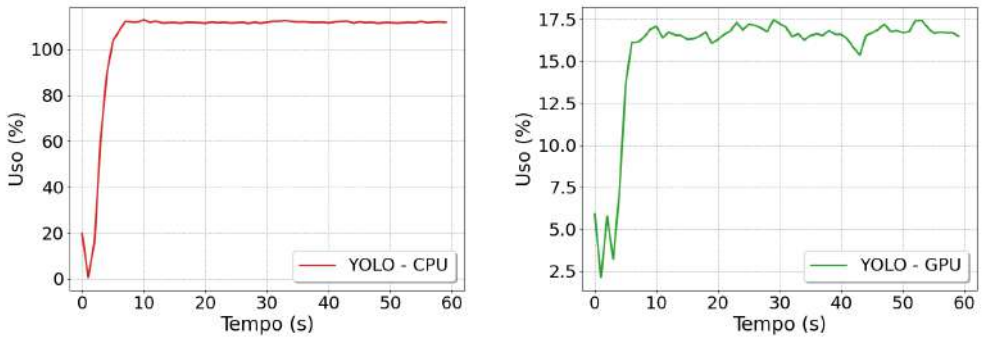
avançada com GPU, a latência é reduzida em cerca de 90%, alcançando valores próximos de 40 ms. Os resultados de vazão também evidenciam a relevância do uso de GPU para essa tarefa: enquanto o uso de CPU não permite atingir os 24 FPS, com GPU é possível superar esse valor, alcançando até 29 FPS.

Analisando a demanda computacional da tarefa de detecção de objetos, a Figura 9a ilustra o uso de CPU para o servidor de borda na configuração básica e de GPU para o servidor na configuração avançada. Na configuração básica, o uso de CPU atinge 113%, ou seja, são utilizados apenas 2 núcleos de processamento dos 8 disponíveis. No entanto, esse baixo uso de CPU está relacionado ao grande número de quadros descartados durante o experimento. Em outras palavras, como os quadros levaram muito tempo para serem processados, a maioria deles foi descartada para não comprometer a fluidez do vídeo. De fato, nossos testes mostram que 89% dos quadros recebidos pelo servidor de borda nesta configuração foram descartados pela aplicação porque quadros mais recentes foram recebidos antes dos quadros que estavam na fila para serem processados. Por outro lado, na configuração avançada, a tarefa de detecção de objeto, que é altamente paralelizável, utilizou 17,5% da GPU. Neste caso, o baixo uso

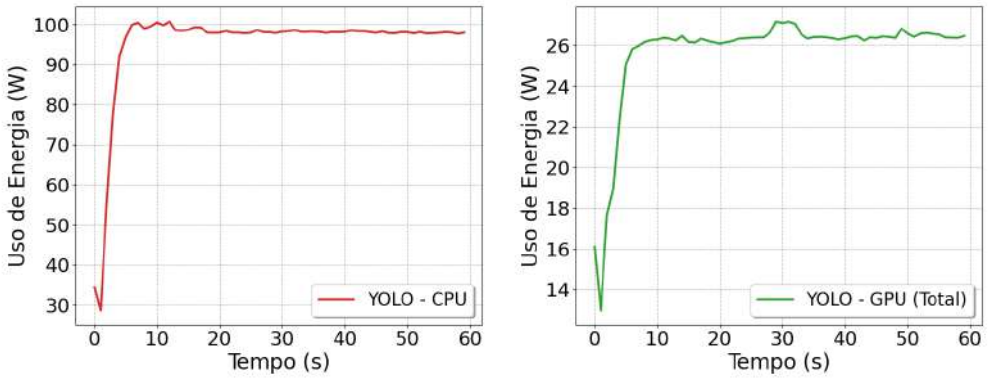
Tabela 2: Vazão Média da Tarefa de Detecção de Objetos Medida em FPS

	TCP		UDP	
Configuração	**Média**	**Desvio Padrão**	**Média**	**Desvio Padrão**
Básica	2	0,02	2	0,05
Avançada	29	1,21	29	0,34

dos recursos de processamento está relacionado com a eficiência da GPU utilizada, uma vez que menos de 1% dos quadros foram descartados neste experimento. O consumo energético, ilustrado na Figura 9b, acompanha, conforme esperado, o uso da CPU e da GPU, atingindo um máximo de 100 W na configuração básica e 27 W na configuração avançada.



(a) Uso de CPU e GPU

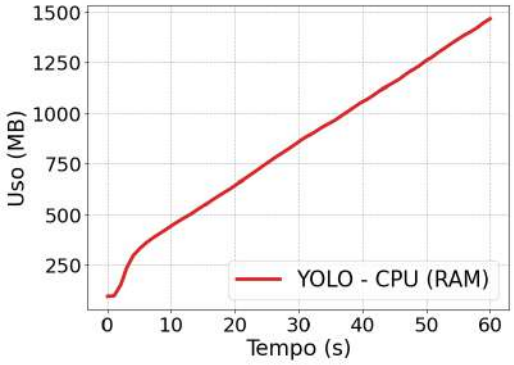


(b) Consumo energético

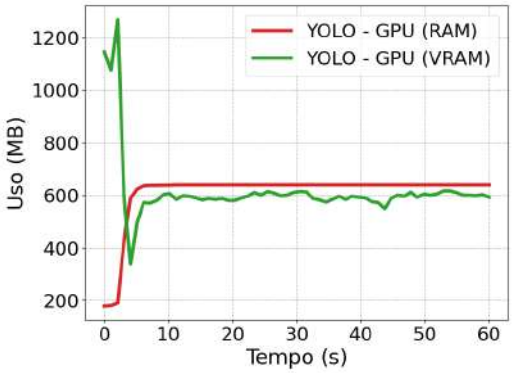
Figura 9: Consumo de recursos de processamento e consumo energético da tarefa de detecção de objetos nas configurações básica e avançada.

O consumo de memória RAM e VRAM para a tarefa de detecção de objetos é apresentado na Figura 10. Na configuração básica, ilustrada na Figura 10a, observa-se um crescimento contínuo no consumo de memória RAM, possivelmente decorrente da não liberação adequada da memória alocada para os quadros descartados. Como o descarte é frequente nesta configuração, muitos dados são mantidos na memória de forma desnecessária. Por outro lado, na configuração avançada, como mostrado na Figura 10b, o consumo de RAM é reduzido para cerca de 639 MB. Isso evidencia como o uso da GPU desloca e divide a carga de trabalho entre a memória RAM e a VRAM. Durante o início do processamento, observa-se um pico de uso da VRAM, alcançando aproximadamente 1200 MB. Esse comportamento decorre do carregamento do modelo e seus

tensores na memória da GPU durante essa fase inicial. Após a inicialização, o consumo de memória estabiliza-se, refletindo a etapa de inferência, na qual o volume de dados processados permanece constante.



(a) Uso de RAM na configuração básica.



(b) Uso de RAM e VRAM na configuração avançada

Figura 10: Uso de memória da tarefa de detecção de objetos nas configurações básica e avançada.

2.4 Conclusão e Trabalhos Futuros

A avaliação de desempenho apresentada permite concluir que a tarefa de SLAM, baseada no ORB-SLAM2, não apresenta benefícios significativos ao ser executada em uma arquitetura heterogênea (CPU e GPU) na borda. Em contrapartida, a tarefa de detecção de objetos, fundamentada em redes neurais, demanda obrigatoriamente a utilização de uma arquitetura heterogênea para seu processamento. Adicionalmente, conclui-se que o protocolo de transporte empregado não exerce impacto relevante no desempenho de ambas as tarefas, sendo, portanto, o TCP o protocolo mais adequado devido à sua confiabilidade. Como etapas futuras, pretende-se utilizar os dados apresentados neste trabalho para a construção de um modelo de carga destinado a aplicações de AR.

3 Transmissão de Vídeo em Redes Sem Fio Utilizando Comunicação Semântica

3.1 Introdução

A crescente demanda por aplicações de transmissão de vídeo em alta resolução, como videoconferências, *cloud gaming* e realidade virtual, impõe desafios significativos à infraestrutura de redes móveis, especialmente no contexto das redes 6G [10, 11]. Essas aplicações exigem não apenas alta largura de banda, mas também baixa latência e alta resiliência à variabilidade das condições de rede.

A comunicação semântica tem sido investigada como uma alternativa promissora para a melhoria da eficiência na transmissão de dados em cenários com restrições de rede. Em vez de transmitir dados brutos, essa abordagem visa transmitir apenas informações com significado relevante, reduzindo o volume de dados a serem enviados [12]. No caso de vídeo, modelos baseados em aprendizado profundo são empregados para extrair elementos semânticos (como objetos, movimentos ou regiões de interesse) e permitir a reconstrução de quadros com qualidade visual satisfatória, mesmo sob restrições de largura de banda [13].

Abordagens recentes, como as baseadas em *autoencoders*, têm demonstrado a capacidade de representar quadros de vídeo de forma compacta, com redução significativa na quantidade de dados transmitidos [13, 14, 15]. No entanto, essas soluções ainda enfrentam limitações na qualidade de reconstrução e na latência em cenários em tempo real. Em contrapartida, os *neural codecs* têm se destacado por integrar a comunicação semântica diretamente aos protocolos de rede, substituindo *codecs* tradicionais por redes neurais treinadas para compressão e reconstrução de vídeo [16, 17, 18, 19].

Dentre essas propostas, o trabalho de Sivaraman *et al.* [16] implementa um sistema de videoconferência funcional sobre WebRTC, utilizando reconstrução neural a partir de *frames* de baixa resolução e um *frame* de referência de alta qualidade. Essa abordagem mostrou-se eficaz mesmo em dispositivos com poder computacional reduzido, mantendo boa fidelidade visual com uso limitado de largura de banda. Entretanto, o modelo foi desenvolvido com foco específico em cenários de webconferência, não apresentando capacidade de generalização ou aplicação eficaz em outros contextos.

Portanto, aplicar comunicação semântica à transmissão de vídeo em tempo real apresenta desafios técnicos. A extração e reconstrução semântica precisa dos quadros é computacionalmente custosa e pode aumentar a latência. Além disso, é necessário garantir consistência semântica entre transmissor e receptor, bem como escalar a solução para múltiplos usuários e tipos de conteúdo. Ainda assim, os trabalhos existentes não fornecem um sistema de transmissão funcional e modular que permita avaliar, de forma sistemática, o impacto da comunicação semântica na qualidade do vídeo e no uso de recursos de rede.

Com base nesses avanços e lacunas, este trabalho propõe o desenvolvimento de um sistema modular para transmissão de vídeo utilizando comunicação semântica capaz de avaliar diferentes abordagens de reconstrução e extração semântica em diferentes cenários.

3.2 Solução Proposta

Este trabalho propõe o desenvolvimento de um *codec* de vídeo semântico modular para transmissão de vídeo em redes sem fio. A proposta é inspirada na arquitetura apresentada

---

O conteúdo deste capítulo foi desenvolvido pelos pesquisadores *Elton Vivot e Kleber Vieira Cardoso.*

em [16], que realiza a reconstrução de quadros de alta resolução a partir de versões em baixa resolução, com suporte de um quadro de referência para aplicar detalhes de textura por meio de uma *pipeline* de super-resolução condicional.

A Figura 11 ilustra a arquitetura geral do sistema proposto. O transmissor captura quadros de vídeo e os transmite em baixa resolução. Periodicamente, um quadro de referência em alta resolução é enviado com o objetivo de permitir a reconstrução de detalhes texturais. No receptor, um modelo de reconstrução (por exemplo, uma *Generative Adversarial Network* (GAN) [20]) realiza a predição do quadro em alta resolução, combinando as informações semânticas extraídas do *frame* de baixa resolução com a referência em alta resolução. No transmissor, detectores semânticos também podem ser utilizados para extrair segmentos do *frame* com base na relevância ou outros critérios. Dessa forma, pode ser feita a transmissão de segmentos com diferentes resoluções ou apenas segmentos que apresentam modificações. Um segmento contendo um texto, por exemplo, pode ser transmitido com resolução maior para garantir a preservação da informação. Segmentos estáticos — ou seja, que não apresentam modificações em relação aos *frames* previamente transmitidos — podem ser omitidos no processo de transmissão.

Para desenvolvimento da solução proposta foi utilizada a biblioteca Python `aiortc`, uma implementação de WebRTC que permite a integração com ferramentas de aprendizado profundo desenvolvidas em `PyTorch` ou `TensorFlow`. A arquitetura modular do sistema permite acoplar diferentes modelos de predição e extração semântica, possibilitando a avaliação de múltiplas abordagens com foco na qualidade da reconstrução e na otimização dos recursos de rede.

A proposta também realiza a coleta de métricas para avaliação da qualidade de serviço e da experiência do usuário. Para métricas visuais, são consideradas as métricas *Peak Signal-to-Noise Ratio* (PSNR), *Structural Similarity Index Measure* (SSIM) e *Learned Perceptual Image Patch Similarity* (LPIPS). Já no âmbito das métricas de rede, são monitoradas a latência de ponta-a-ponta, a taxa de perda de pacotes e a largura de banda efetivamente utilizada.

A arquitetura modular adotada visa à construção de um sistema flexível e adaptável, capaz de integrar diversas ferramentas voltadas à comunicação semântica. Essa modularidade permite a substituição e combinação de diversos componentes, como detectores semânticos, modelos de reconstrução e codificadores, viabilizando a experimentação e a análise comparativa de distintas abordagens. A análise é realizada com base em métricas visuais e de rede, coletadas de forma integrada, de modo a permitir uma validação científica sistemática das alternativas implementadas.

3.3 Resultados Preliminares

A solução proposta encontra-se atualmente em processo de desenvolvimento. Até o momento, foram implementados os principais componentes do sistema, incluindo:

- Transmissão de *frames* em baixa resolução utilizando WebRTC com a biblioteca `aiortc`;
- Integração de ferramentas semânticas (como GANs) no transmissor e no receptor para reconstrução dos quadros;
- Coleta de métricas de qualidade visual e métricas de rede, a partir dos quadros recebidos.

Também foram estudadas diferentes arquiteturas de redes neurais generativas para integrar ao receptor do *codec* semântico. O objetivo da análise foi selecionar uma alternativa adequada para melhorar a resolução de quadros transmitidos em baixa qualidade. Foram avaliadas as seguintes arquiteturas: GAN, *Deep Convolutional Generative Adversarial Network* (DCGAN),

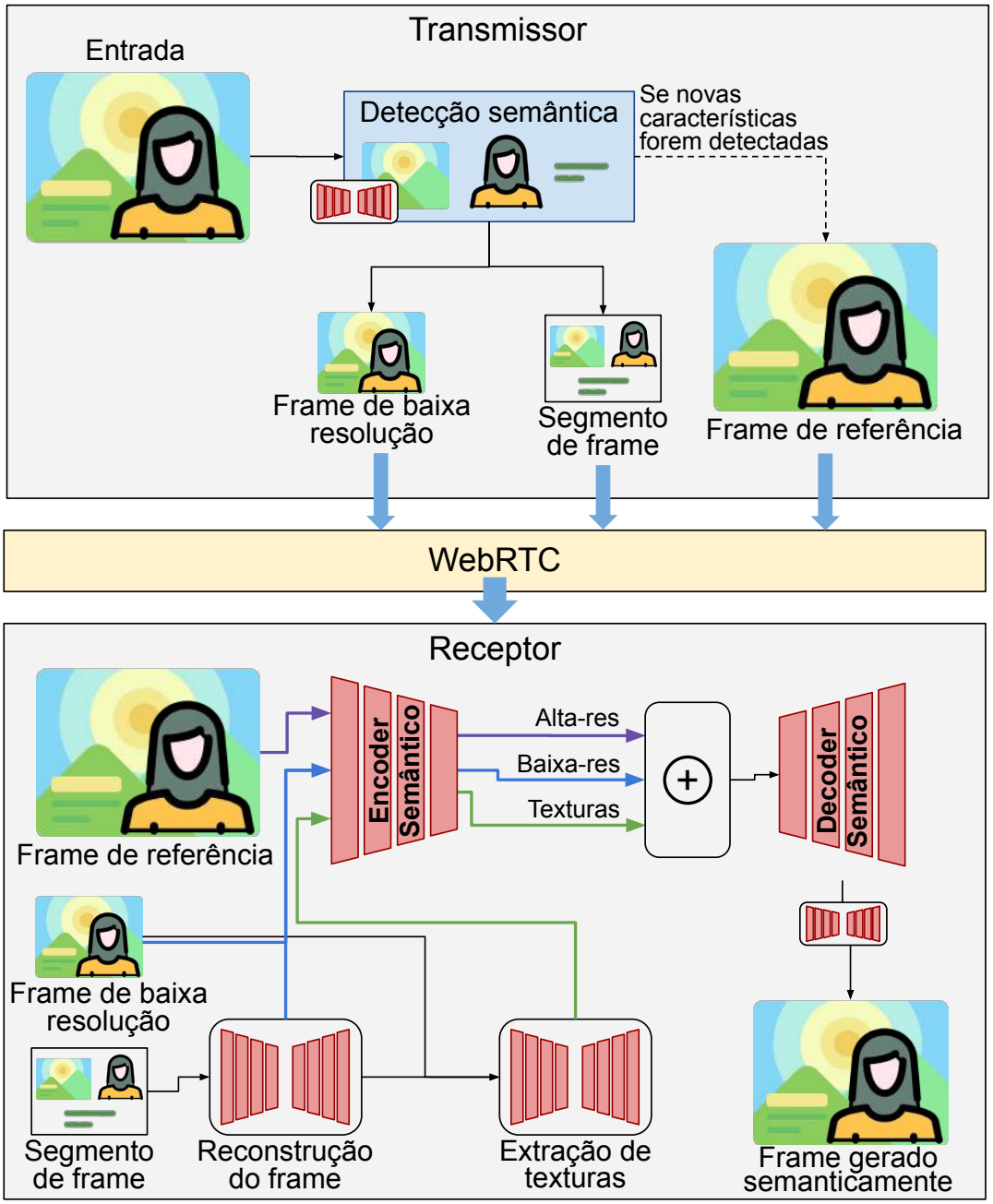


Figura 11: Arquitetura do *codec* de vídeo semântico proposto.

Context Encoder, *Boundary Equilibrium Generative Adversarial Network* (BEGAN), *Super-Resolution Generative Adversarial Network* (SRGAN) e *Enhanced Super-Resolution Generative Adversarial Network* (ESRGAN). As GANs tradicionais e o DCGAN foram considerados para fins comparativos, por apresentarem estruturas mais simples com redes densas ou convolucionais. O *Context Encoder* foi analisado por sua capacidade de preencher regiões faltantes com base em contexto visual. A BEGAN utiliza um discriminador baseado em *autoencoders* e uma métrica de convergência para regular o treinamento. Já as redes SRGAN e ESRGAN aplicam técnicas de super-resolução guiadas por perdas perceptuais baseadas em VGG19, sendo capazes de reconstruir imagens de alta resolução a partir de versões degradadas. Devido à sua relação

direta com o problema de reconstrução de vídeo, as redes de super-resolução foram selecionadas como candidatas iniciais para testes. No entanto, a ESRGAN, embora apresente melhores resultados visuais, foi identificada como altamente demandante em termos de recursos computacionais, o que pode limitar sua aplicabilidade em tempo real. Como resultado, a SRGAN foi escolhida como opção inicial para integração ao sistema, por apresentar bom equilíbrio entre qualidade e custo computacional.

3.4 Trabalhos Futuros

Os próximos passos deste trabalho estão direcionados à consolidação da arquitetura proposta como uma plataforma experimental para avaliação de técnicas de comunicação semântica aplicadas à transmissão de vídeo. As etapas a seguir estão planejadas:

- Automatizar a coleta, armazenamento e visualização gráfica das métricas de rede (como latência, largura de banda e perda de pacotes) e das métricas visuais (como PSNR, SSIM e LPIPS), permitindo a análise em tempo real do impacto das condições de rede na qualidade dos *frames* reconstruídos;
- Integrar e comparar diferentes ferramentas semânticas para detecção e reconstrução de conteúdo, visando validar a modularidade do sistema e identificar combinações mais eficazes em diferentes cenários de transmissão;
- Incorporar técnicas de segmentação semântica para priorização de regiões de interesse nos quadros de vídeo, possibilitando a transmissão seletiva de segmentos com base em critérios como movimento, presença de texto ou importância semântica;
- Realizar experimentos controlados com simulações de degradação de rede (por exemplo, variações de perda de pacotes e limitação de banda) para avaliar a robustez das soluções integradas e sua capacidade de adaptação;
- Testar diferentes configurações de resolução na entrada e na saída dos modelos de reconstrução, com o objetivo de encontrar um equilíbrio entre eficiência de compressão e qualidade visual percebida;
- Comparar a solução proposta com *codecs* tradicionais como VP8 e VP9 em termos de desempenho visual, utilização de rede e resiliência a falhas de transmissão.

Essas etapas buscam quantificar o impacto da comunicação semântica na qualidade visual dos vídeos e na utilização da rede, apoiando o desenvolvimento de soluções mais eficientes para redes com restrições de capacidade.

4 VR-GX: Um modelo de alocação de recursos baseado em QoE com atenção para jogos de realidade virtual em nuvem

4.1 Introdução

As redes 6G prometem transformar a conectividade ao oferecer suporte a ambientes densos e comunicações ultra confiáveis. Entre as tecnologias habilitadoras estão a *Semantic Communication* (SC) e a *Computing and Network Convergence* (CNC) [21]. A SC busca reduzir a carga de transmissão focando no significado da informação, enquanto a CNC aproxima os *Computing Node*s (CNs) dos usuários, integrando recursos de rede e computação para lidar com aplicações intensivas e sensíveis à latência.

Uma das principais aplicações imersivas definidas pelo 3GPP é o VR-CG [22, 23, 24], que viabiliza a execução de jogos em alta fidelidade renderizados remotamente por CNs na infraestrutura da *Radio Access Network* (RAN), reduzindo a exigência de *hardware* avançado nos dispositivos dos usuários [25, 26]. No entanto, o VR-CG impõe desafios à alocação de recursos, devido à intensa competição por capacidade de transmissão e roteamento gerada por aplicações imersivas. As soluções atualmente propostas para esse problema [23, 25, 26] frequentemente negligenciam aspectos essenciais, como modelos realistas de comunicação sem fio, seleção dinâmica de resolução, comunicação semântica e FPS adaptativa. Dessa forma, uma formulação abrangente torna-se fundamental para enfrentar os desafios práticos do VR-CG [23]. Este trabalho, portanto, propõe uma formulação matemática para o problema de alocação de recursos no contexto do VR-CG, contemplando todas as definições estabelecidas pelo 3GPP e visando preencher as lacunas identificadas na literatura.

4.2 Modelo de Sistema

O sistema RAN considerado foi modelado como um grafo $G = \{V, E\}$, onde $V = \{c_0\} \cup \mathcal{T} \cup \mathcal{B} \cup \mathcal{C}$ representa os nós da rede, incluindo o núcleo da rede c_0, os nós de transporte $\mathcal{T}$, as *Base Station*s (BSs) $\mathcal{B}$ e os CNs $\mathcal{C}$. Cada CN $c_m \in \mathcal{C}$ possui capacidades de processamento, memória, rede e renderização, representadas por c_m^{CPU}, c_m^{Mem}, c_m^{Net} e c_m^{GPU}, respectivamente. Os enlaces $e_{ij} \in E$ conectam os nós com capacidade e_{ij}^{Cap} e latência e_{ij}^{Lat}. Além disso, consideramos o ambiente VR-CG composto por um conjunto de usuários $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ conectados as BSs via enlaces sem fio. Cada usuário u_i possui uma *Signal-to-Interference-plus-Noise Ratio* (SINR) e um dispositivo HMD com resolução máxima $Res(u_i)$ e taxa de quadros máxima $FPS(u_i)$.

Adicionalmente, propôs-se um ambiente dinâmico com alocação de recursos sensível à atenção, no qual cada usuário observa um conjunto de objetos virtuais $\mathcal{O}_i$ em seu *Field of View* (FoV). Esses objetos são renderizados a partir de um conjunto de resoluções possíveis $\mathcal{R}$ e transmitidos considerando um conjunto de FPS $\mathcal{F}$. A resolução de cada objeto depende do nível de atenção $\lambda_i^j \in \mathbb{R}$ do usuário u_i sobre o objeto $o_j \in \mathcal{O}_i$, reduzindo a carga de transmissão ao priorizar objetos de maior interesse.

O conteúdo desta seção consiste em um resumo adaptado pelo pesquisador *Kleber V. Cardoso*, com base no artigo publicado por Gabriel M. Almeida, João Paulo Esper, Luiz A. DaSilva e Kleber V. Cardoso: "*VR-GX: an Attention-aware QoE-based resource allocation model for VR-Cloud Gaming*", nos Anais do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Natal/RN, 2025, pp. 616-629, doi: https://doi.org/10.5753/sbrc.2025.6328.

4.2.1 Formulação do Problema

Esta seção apresenta a formulação do problema *Virtual Reality cloud Gaming Resource Allocation Based on Quality of Experience* (VR-GX), alinhada aos padrões do 3GPP [22, 24, 23]. O objetivo é selecionar a resolução de renderização dos objetos virtuais e a taxa de quadros apropriada para cada usuário, maximizando a QoE sob restrições de capacidade da infraestrutura.

Para modelar os recursos de comunicação, definem-se as seguintes variáveis: $y_l^i \in \{0,1\}$, que indica se o usuário u_i é admitido pela BS b_l, e $x_l^i \in \mathbb{Z}$, que representa a largura de banda alocada. Para os recursos computacionais, consideram-se: $w_{i,j}^k \in \{0,1\}$ para selecionar a resolução r_k do objeto o_j visto pelo usuário u_i; $v_i^f \in \{0,1\}$ para a taxa de quadros f; e $z_i^m \in \{0,1\}$ para indicar se a aplicação VR-CG do usuário u_i será executada no nó de computação c_m.

O modelo de QoE considera o nível de atenção do usuário aos objetos localizados dentro de FoV, a resolução desses objetos e a taxa de quadros. A formulação baseia-se na lei de satisfação de Weber-Fechner, conforme descrito em [27], definindo a QoE do usuário u_i como:

$$\zeta(i) = \sum_{f \in \mathcal{F}} v_i^f \ln(f) + \sum_{o_j \in \mathcal{O}_i} \sum_{r_k \in \mathcal{R}} \left(w_{i,j}^k \ln(\tau(j,k)\lambda_i^j) \right), \tag{1}$$

em que $\tau(j,k)$ é o coeficiente de qualidade da resolução e λ_i^j é o nível de atenção do usuário u_i ao objeto o_j.

A função objetivo busca maximizar a soma da QoE dos usuários, enquanto respeita as restrições de capacidade dos elementos de rede e de computação, e requisitos de comunicação como latência e vazão. O modelo de otimização é definido da seguinte forma:

$$\text{maximizar} \sum_{u_i \in \mathcal{U}} \zeta(i). \tag{2}$$

Sujeito à:

$$\sum_{r_k \in \mathcal{R}} w_{i,j}^k = 1, \qquad \forall u_i \in \mathcal{U},\ o_j \in \mathcal{O}_i. \tag{3}$$

$$\sum_{f \in \mathcal{F}} v_i^f = 1, \qquad \forall u_i \in \mathcal{U}. \tag{4}$$

$$\sum_{u_i \in \mathcal{U}} z_i^m \mathbf{G}(\mathcal{O}_i, \mathbf{w}_{i,j}^k) \leq c_m^{GPU}, \qquad \forall c_m \in \mathcal{C}, \tag{5}$$

$$\sum_{o_j \in \mathcal{O}_i} \sum_{r_k \in \mathcal{R}} \left(w_{i,j}^k \Psi(j,k) \mathbf{C}(\mathcal{O}_i) \right) \leq \sum_{s \in \mathbb{S}} \sum_{b_l \in \mathcal{B}} x_l^i \log_2 \left(1 + SINR(u_i, s)\right) \tag{6}$$

$$\mathbf{P}(i) + \mathbf{T}(i) + \mathbf{C}(i) + \mathbf{Q}(i) + \mathbf{R}(i) + \mathbf{M}(i) \leq E_i, \quad \forall u_i \in \mathcal{U}. \tag{7}$$

As Equações (3)-(4) definem que deve ser selecionada exatamente uma resolução por objeto virtual e uma taxa de atualização de quadros para cada usuário. A Equação (5) define a restrição de capacidade de renderização dos nós de computação, onde $\mathbf{G}$ representa a carga de renderização (*flops/pixel*) de acordo com as resoluções escolhidas para os objetos ($\mathbf{w}_{i,j}^k$). A Equação (6) define que a vazão oferecida ao usuário deve ser maior do que sua demanda, onde $\Psi(j,k)$ representa o tamanho da imagem transmitida, $\mathbf{C}(\mathcal{O}_i)$ representa a taxa de compressão

de dados do *codex* de vídeo e $SINR(u_s, s)$ representa a SINR do usuário para o canal s da BS a qual está conectado. Por fim, a Equação (7) representa a restrição de latência fim-a-fim, garantindo que a transmissão dos quadros de vídeo renderizados pelo nó de processamento sejam entregues obedecendo a latência máxima de geração de quadros (E_i). Para o cálculo da latência fim a fim, são consideradas as seguintes componentes: latência de propagação ($\mathbf{P}(i)$), transmissão ($\mathbf{T}(i)$), processamento ($\mathbf{C}(i)$), fila ($\mathbf{Q}(i)$), roteamento ($\mathbf{R}(i)$) e renderização ($\mathbf{M}(i)$).

A formulação do problema VR-GX caracteriza-se como sendo NP-difícil, classificada como um problema *Mixed-Integer Nonlinear Programming* (MINLP), contendo variáveis binárias e relações não lineares. Tais características impõem a necessidade de abordagens heurísticas para a obtenção de soluções eficientes. A resolução exata do problema, por meio de ferramentas como CPLEX e Gurobi, torna-se inviável em cenários práticos, devido ao elevado tempo computacional requerido. Com o intuito de contornar essa limitação, propõe-se uma heurística evolutiva determinística de natureza gulosa, fundamentada na formulação matemática previamente apresentada. Tal abordagem permite encontrar soluções com elevados níveis de QoE de forma eficiente, mesmo em cenários práticos compostos por um grande número de usuários e BSs.

4.3 Resultados Simulados

A formulação do problema VR-GX foi avaliada por meio da comparação entre as soluções ótimas, as soluções aproximadas obtidas pela heurística proposta e os resultados do modelo *Rendering Capacity Allocator* (RCA), conforme apresentado em [27]. A simulação segue o ambiente descrito em [28], com 475 BSs distribuídas em uma área de 4 km^2 (2 km $\times$ 2 km), cada uma com 100 MHz de banda. Foram selecionadas quatro BSs com melhor qualidade de canal para compor o cenário de avaliação. O conjunto de dados referente à atenção dos usuários está disponível publicamente, assim como o código-fonte desenvolvido neste trabalho. As simulações foram realizadas em Python 3.10.12 com *CPLEX* (v22.11) e *docplex*, em um sistema com processador Intel® Core™ i7-1255U e 32 GB de RAM.

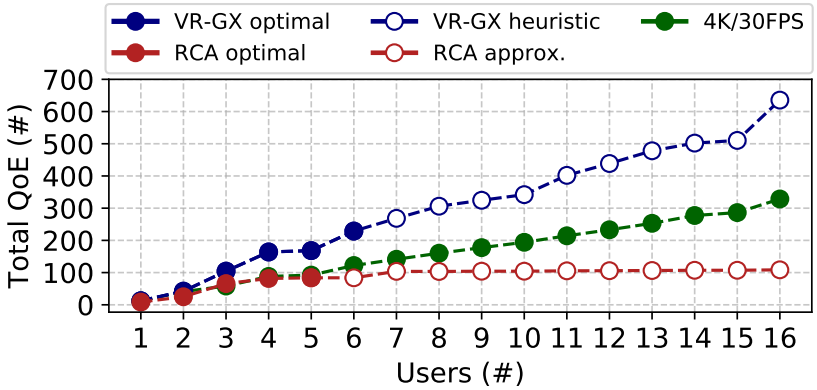


Figura 12: QoE total.

A Figura 12 ilustra o QoE total obtido *versus* o número de usuários admitidos. A partir dos resultados verifica-se que o modelo VR-GX supera o RCA e a alocação fixa (4K/30 FPS) em termos de QoE total e médio. Essa vantagem ocorre por considerar FPS na métrica de

https://github.com/HongyangDu/AttentionQoE
https://github.com/LABORA-INF-UFG/paper-GJLK-2025

QoE, conforme padronizado pelo 3GPP [22, 24, 23]. Em cenários com mais usuários, o RCA prioriza usuários com melhor canal, o que reduz a equidade e a QoE média. A heurística VR-GX apresenta desempenho próximo ao ótimo, mantendo altos níveis de QoE e equidade entre usuários.

Na Figura 13, observa-se que o VR-GX, tanto em sua forma ótima quanto na versão heurística, admite todos os usuários enquanto mantém níveis mais elevados de equidade em termos de QoE e latência, conforme mensurado pelo índice de equidade de Raj Jain. Em contraste, o modelo RCA prioriza a latência e tende a favorecer usuários com melhores condições de canal, o que compromete a equidade e a QoE em cenários mais carregados.

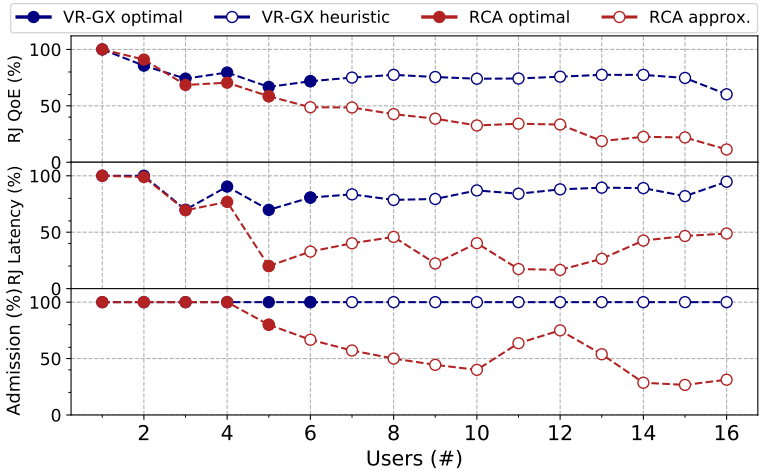


Figura 13: Comparação de equidade em QoE e latência, e número de usuários admitidos.

Por fim, a Figura 14 demonstra que o modelo RCA consome mais largura de banda, alocando recursos em excesso na tentativa de maximizar o *throughput*. Em contrapartida, o VR-GX satisfaz apenas os limites necessários de comunicação, evitando o sobreprovisionamento de recursos. Já a Figura 15 evidencia que a heurística proposta é capaz de encontrar soluções de alta qualidade em menos de 4 ms, mostrando-se adequada para aplicações práticas em tempo real, ao passo que os modelos baseados em soluções ótimas demandam tempos computacionais significativamente maiores.

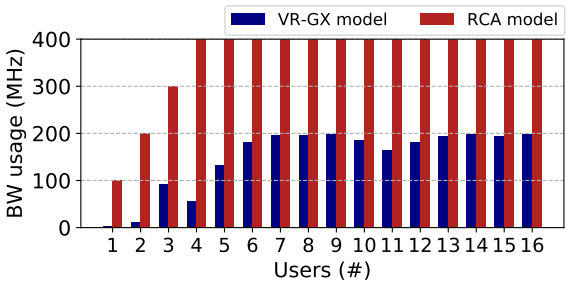


Figura 14: Uso de banda por abordagem.

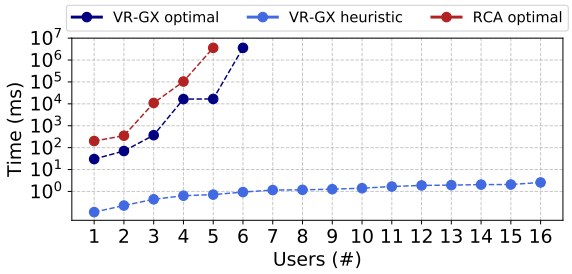


Figura 15: Tempo de execução por abordagem.

4.4 Conclusão

Este estudo apresenta uma nova formulação, denominada VR-GX, que aborda de forma eficaz os desafios associados à alocação de recursos em aplicações de VR-CG, oferecendo soluções superiores em termos de QoE e equidade de latência, quando comparadas às abordagens existentes. O algoritmo heurístico proposto proporciona uma alternativa prática para lidar com a complexidade das demandas dos usuários e as restrições da rede, demonstrando-se adequado para aplicações em tempo real com grande número de usuários. Como direções futuras, pretende-se refinar a heurística desenvolvida, explorar abordagens baseadas em aprendizado de máquina para alocação dinâmica de recursos, realizar experimentos práticos para coleta de avaliações subjetivas (*Mean Opinion Score*), com o objetivo de validar o modelo de QoE proposto, além de estender o *framework* para contemplar a mobilidade dos usuários e condições dinâmicas da rede.

5 Transferência de Energia sem Fio com Alto Rendimento e Segura

5.1 Introdução

O avanço das comunicações sem fio leva a uma sociedade cada vez mais digitalizada, orientada por dados e autônoma. Infelizmente, o carregamento com fio e a necessidade de substituição de baterias ainda são barreiras críticas para uma conectividade móvel ilimitada, escalável e sustentável. O carregamento com fio onipresente geralmente é proibitivamente caro, especialmente em implantações industriais da IoT, enquanto soluções baseadas em bateria enfrentam desafios como vida útil limitada e a necessidade de substituição, o que não é econômico nem ecologicamente sustentável.

A indústria e a academia concordam que as abordagens de *Energy Harvesting* (EH) e *Wireless Power Transfer* (WPT) podem em breve superar essa barreira [29]. Notavelmente, a WPT pode ser inevitavelmente necessária em aplicações com requisitos de *Quality of Service* (QoS), pois EH a partir da energia ambiente é não determinística e dependente da localização, muitas vezes exigindo transdutores de tamanho relativamente grande.

As soluções de WPT amplamente disponíveis no mercado dependem de técnicas reativas de curto alcance, explorando campos magnéticos ou elétricos, com suporte limitado a múltiplos usuários, ainda restringindo a manobrabilidade, mobilidade e escalabilidade dos dispositivos carregados. Por outro lado, a tecnologia de RFWPT baseada em radiação suporta nativamente o carregamento de múltiplos usuários e permite raios de carregamento maiores [29]. No entanto, a baixa *Power Transfer Efficiency* (PTE) de ponta a ponta inerente ao carregamento sem fio por *Radio Frequency* (RF) e as preocupações relacionadas à segurança são obstáculos críticos que retardam as atividades de padronização.

Esses desafios têm motivado, principalmente, aplicações personalizadas de carregamento de IoT de baixa potência, confiando em: i) *Energy Beamforming* (EBF) e otimização de forma de onda [30, 31, 32, 33]; ii) sistemas distribuídos e massivos de antenas [32]; iii) matrizes refletoras inteligentes e metasuperfícies reconfiguráveis [34, 33]; e iv) *Energy Transmitter* (ET) flexíveis [29] (ou seja, ETs móveis e/ou equipados com antenas rotativas); enquanto, em muitos casos, incorporam explicitamente o controle da exposição à radiação do *Electromagnetic Field* (EMF) [30, 32].

Considera-se que uma combinação abrangente e a otimização holística dessas, bem como de outras tecnologias promissoras, são fundamentais para tornar a PTE dos sistemas RFWPT competitiva, viabilizando sua aplicação em cenários que demandam carregamento de potência relativamente elevada. Nesse contexto, algumas *startups*, como a Energous, a Ossia e a GuRu, têm desenvolvido soluções baseadas em RFWPT voltadas a aplicações com requisitos energéticos mais exigentes.

Embora RFWPT esteja avançando rapidamente em direção à adoção comercial, sua implementação e operação ainda requerem o desenvolvimento de estratégias e protocolos cautelosos. Neste trabalho, essa problemática é analisada em profundidade, sendo apresentadas as seguintes contribuições principais:

O conteúdo desta seção consiste em um resumo adaptado pelo pesquisador *Richard Demo Souza*, com base no artigo publicado por O. L. A. López et al., "*High-Power and Safe RF Wireless Charging: Cautious Deployment and Operation*," in IEEE Wireless Communications, vol. 31, no. 6, pp. 118-125, December 2024, doi: 10.1109/MWC.017.2300462.

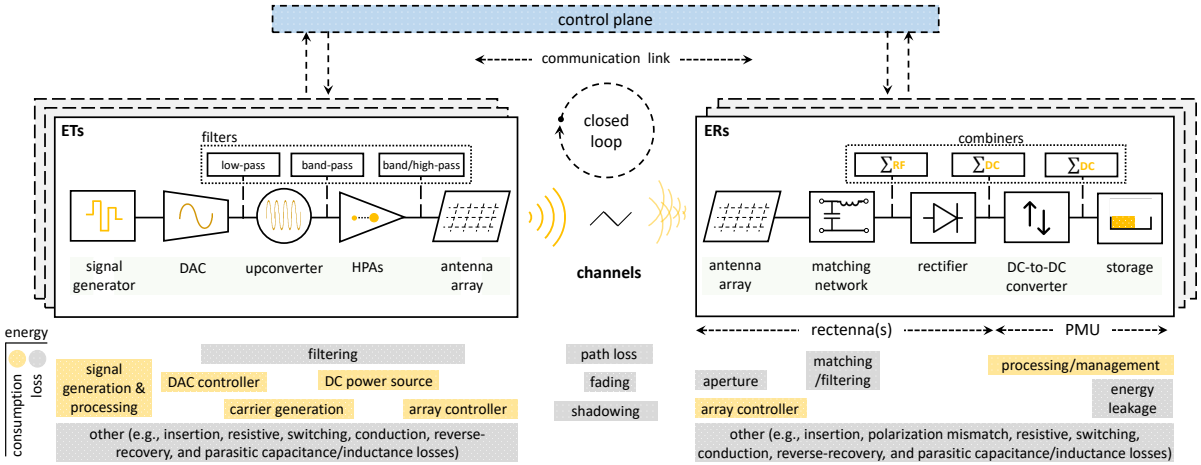


Figura 16: Diagrama de blocos da arquitetura de um sistema RFWPT e principais fontes de consumo/perda de energia.

- Apresentam-se os principais fatores que afetam a PTE dos sistemas RFWPT e a necessidade de otimização conjunta.
- Revisam-se métricas de exposição ao EMF e limites de segurança, destacando que o carregamento em campo próximo pode reduzir significativamente a exposição.
- Propõe-se um sistema ciberfísico para WPT eficiente e segura, ilustrando a PTE de arquiteturas MIMO modernas e o potencial de RIS na gerenciabilidade da radiação EMF.
- Identificam-se os principais desafios e direções futuras de pesquisa.

5.2 RFWPT

A arquitetura de um sistema RFWPT é ilustrada na Figura 16 e consiste em: i) ETs; ii) canais sem fio; e iii) *Energy Receivers* (ERs). O sinal de carregamento é gerado no domínio digital e convertido para analógico pelo conversor digital-analógico, seguido por conversão de frequência para o domínio RF, amplificação e transmissão via antenas. No receptor, o sinal recebido passa por uma rede de retificação, formando as chamadas *rectennas*. Para maximizar a transferência de potência, é necessário um circuito de adaptação entre a antena e a rede de retificação. O sinal retificado é processado por uma unidade de gerenciamento de energia, podendo incluir conversores e armazenamento de energia. Dois importantes desafios precisam ser resolvidos para viabilizar a tecnologia RFWPT, como segue.

PTE Fim-a-Fim: O carregamento por RFWPT é mais lento e menos eficiente do que carregadores com fio ou tecnologias de carregamento indutivo devido às múltiplas fontes de consumo e perda de energia. As principais perdas de energia ocorrem em: processamento de sinal de banda base, geração de portadora e amplificação no ET; processamento de sinal e gerenciamento de energia no ER; filtros de remoção de componentes de alta frequência e harmônicos; atenuação devido a perdas de canal, desvanecimento e obstruções; ineficiências na conversão de energia da antena receptora.

A eficiência de cada bloco do sistema impacta, de forma multiplicativa, a eficiência global da transferência de energia. Nesse contexto, melhorias isoladas e não coordenadas podem resultar

em soluções subótimas. Alguns exemplos de *trade-offs* são: reduzir a resolução do conversor analógico digital diminui o consumo de energia, mas aumenta a necessidade de filtragem; aumentar o número de subportadoras melhora a eficiência da *rectenna*, mas eleva o consumo computacional; reduzir as cadeias de RF reduz o consumo no transmissor, mas limita os ganhos de *beamforming* e a capacidade de carregar múltiplos dispositivos simultaneamente.

Para máxima otimização, um sistema RFWPT deve operar em malha fechada, necessitando de um canal de comunicação reverso para aquisição de informação do estado do canal.

Exposição a EMF: As transmissões de RF são não ionizantes, mas há preocupações com efeitos biológicos indiretos. A Agência Internacional de Pesquisa sobre o Câncer classificou as emissões de RF como possivelmente cancerígenas (Grupo 2B). Embora não haja comprovação definitiva, há regulamentações para limitar a exposição. Organizações como *Federal Communications Commission* (FCC), União Internacional de Telecomunicações (UIT) e *International Commission on Non-Ionizing Radiation Protection* (ICNIRP) estabelecem limites de exposição baseados em métricas como taxa de absorção específica para frequências < 6 GHz, e densidade de potência absorvida e densidade de energia absorvida para frequências > 6 GHz.

Os limites são definidos com margens conservadoras para garantir segurança. O carregamento RF de alta potência é viável apenas em curtas distâncias devido às perdas de canal. Para evitar exposição excessiva, é necessário *beamforming* preciso, especialmente em altas frequências e na região de campo próximo. Resultados indicam que, sob condições de campo próximo, a exposição EMF pode ser mantida dentro dos limites seguros, tornando viável o carregamento de dispositivos portáteis sem riscos significativos para os usuários.

Por fim, estratégias cautelosas devem ser implementadas para garantir transparência na exposição EMF, permitindo a adoção segura do RFWPT.

5.3 Proposta de um Sistema de Carregamento Sem Fio Eficiente e Seguro

Os principais componentes e a visão para um sistema ciberfísico RFWPT em ambientes internos são ilustrados na Figura 17. Um design holístico adequado desses componentes é essencial para maximizar o potencial de RFWPT, incentivando padronizações e produtos comerciais.

Transmissores e Refletores Inteligentes: A implementação de MIMO massivo pode mitigar perdas no canal e melhorar a eficiência energética [29, 32]. No entanto, seu alto custo motiva pesquisas em arquiteturas MIMO de baixo consumo, como sistemas híbridos analógico-digitais e RIS, que permitem direcionar sinais sem consumo adicional de RF. Alternativas ativas de RIS oferecem ganhos de desempenho, mas com maior consumo energético e complexidade na aquisição de informação sobre o estado do canal.

Receptores de Energia: Os circuitos dos ERs devem ser otimizados para eficiência em níveis elevados de potência de RF. Retificadores multiestágio com diodos Schottky são adequados para esse fim, assim como capacitores de baixa resistência para minimizar perdas. Estratégias como aumento do número de antenas receptoras e utilização de bandas largas podem melhorar a captação de energia. A otimização do design dos ERs deve equilibrar eficiência, custo e conformidade com limites de exposição EMF. Metamateriais podem permitir ERs compactos e eficientes, mas podem exigir separação entre funções de WPT e transmissão de dados sem fio.

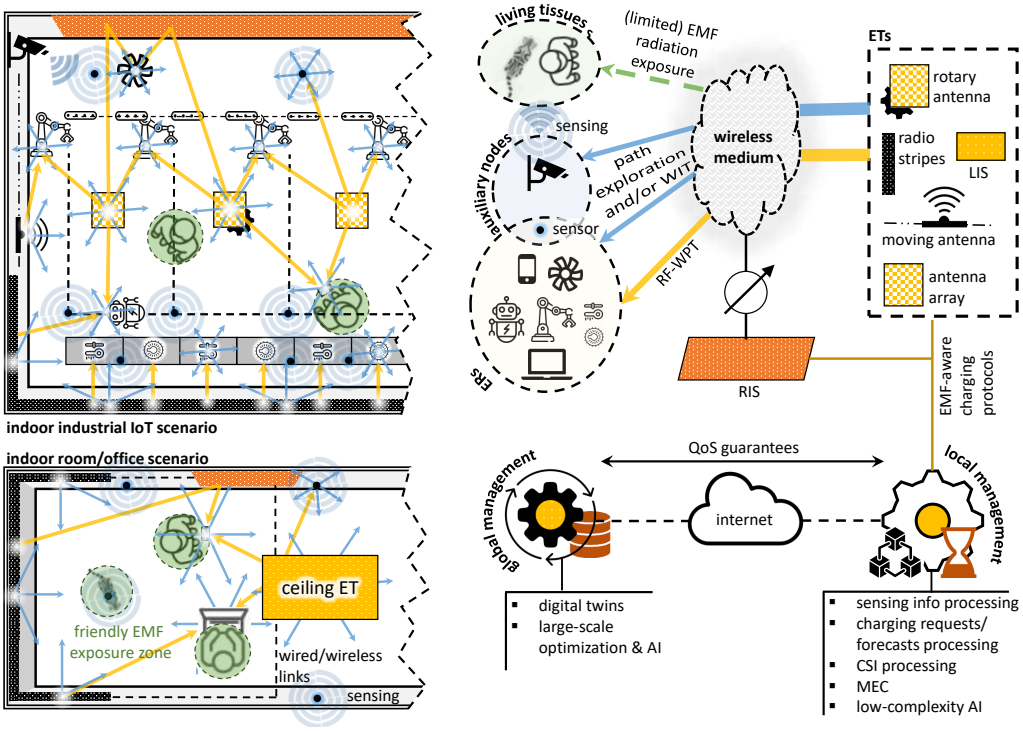


Figura 17: Visão para um sistema RFWPT competitivo e dois cenários-chave: sala/escritório e IoT industrial.

Nós Auxiliares: Sensores térmicos, radares e câmeras podem auxiliar na detecção de ERs e obstáculos, garantindo conformidade com limites de exposição EMF. Serviços de localização aprimorados permitem ajustar a potência do RFWPT para evitar riscos à saúde. Sensores de RF podem ainda mapear a energia disponível, otimizando o gerenciamento da carga.

Protocolos de Carregamento e Gerenciamento: Os protocolos de carregamento envolvem troca de informações entre transmissores e receptores para aquisição de *Channel State Information* (CSI), requisição de carga e otimização do consumo energético. Infraestruturas de sensoriamento mais sofisticadas permitem carregamento seguro na presença de humanos ao monitorar a exposição EMF. O sistema de gerenciamento de RFWPT deve equilibrar eficiência energética e segurança, combinando otimização local e global. O gerenciamento local lida com tarefas de complexidade moderada, enquanto o gerenciamento global emprega inteligência artificial e gêmeos digitais para otimizações avançadas. O sistema pode coexistir, cooperar ou ser co-projetado com redes de comunicação sem fio, sendo a última opção a mais avançada.

5.4 Conclusão

Apresenta-se, neste trabalho, uma visão de um sistema ciberfísico voltado ao carregamento por RF de alta potência eficiente e seguro. São discutidos os principais fatores que influenciam a PTE de ponta a ponta, bem como as métricas de exposição ao EMF e os respectivos limites de segurança. Por fim, ressalta-se a necessidade do desenvolvimento de arquiteturas e protocolos de carregamento que maximizem a PTE, ao mesmo tempo em que assegurem conformidade com as regulamentações de exposição ao EMF.

6 Sistema de Posicionamento BLE Usando Fusão de AoA e RSSI

6.1 Introdução

Sistemas de localização *indoor* são essenciais para fornecer rastreamento e localização precisos em ambientes onde os sinais tradicionais de GNSS não estão disponíveis ou são pouco confiáveis, como em interiores de edifícios [35]. Esses sistemas são particularmente valiosos em indústrias como varejo, saúde, manufatura e logística, onde o rastreamento preciso melhora a eficiência operacional, a segurança e a experiência do cliente [36, 37].

Apesar de sua utilidade, métodos comuns de localização *indoor*, como RSSI, AoA, *Time of Arrival* (ToA) e *Time Difference of Arrival* (TDoA), enfrentam desafios significativos, incluindo interferência de sinal e efeitos de multipercurso, que degradam a precisão em ambientes reais [38]. Além disso, há um *trade-off* entre consumo de energia e desempenho do sistema, especialmente para dispositivos alimentados por bateria, o que afeta a usabilidade e a satisfação do usuário [39].

Para enfrentar esses desafios, propõe-se um método híbrido de localização baseado em BLE, que combina medidas de RSSI e AoA utilizando uma técnica de fusão de sensores baseada no *Kalman Filter* (KF) [40]. Ao integrar múltiplos métodos de posicionamento, como multilateração e triangulação, a abordagem proposta melhora a precisão em ambientes internos onde distorções de sinal e interferências são comuns.

Este trabalho é validado utilizando o conjunto de dados de Girolami et al. [41], que inclui medições de RSSI e AoA coletadas em diversos cenários. O desempenho do método proposto, denominado *Angle - Received Signal Strength Indicator Fusion Localization* (ARFL), foi comparado com abordagens tradicionais baseadas exclusivamente em RSSI ou AoA, evidenciando a superioridade do ARFL sob várias métricas de erro.

Diante desse contexto, este estudo avalia a influência da fusão de sensores na precisão de um sistema de localização *indoor* baseado em RSSI e AoA. Para isso, são empregados múltiplos KFs nos conjuntos de dados de medições de RSSI e AoA de [41], cujas saídas são combinadas por meio de técnicas de fusão sensorial. A principal contribuição é uma nova variação da técnica *Track-to-Track Fusion* (T2TF), que integra filtros de Kalman aplicados a medições BLE de AoA e RSSI. Diferentemente de abordagens convencionais, que combinam medições de redes distintas, propõe-se a fusão de informações extraídas de um único sinal recebido. Optou-se pelo uso de multilateração, trigonometria com RSSI e AoA, e triangulação, devido à sua aplicabilidade em tempo real. Esses métodos dispensam a criação de bancos de dados de *fingerprinting*, facilitando a adaptação a mudanças ambientais sem exigir treinamento extenso ou *hardware* avançado.

6.2 Solução Proposta

Este trabalho considera uma sala de 112m^2, onde a *Region of Interest* (RoI) é retangular, com dimensões de 12×6 metros. A RoI contém quatro âncoras, com a n-ésima âncora posicionada em coordenadas conhecidas $\begin{bmatrix} x_n & y_n \end{bmatrix}^T \in$ RoI, para $n \in \{1, \ldots, 4\}$. O objetivo é estimar

O conteúdo desta seção consiste em um resumo adaptado pelo pesquisador *Richard Demo Souza*, com base no artigo publicado por A. Fabris, O. K. Rayel, J. L. Rebelatto, G. L. Moritz and R. D. Souza, "AoA and RSSI-Based BLE Indoor Positioning System With Kalman Filter and Data Fusion," in IEEE Internet of Things Journal, aceito para publicação, 2025, doi: 10.1109/JIOT.2025.3530866.

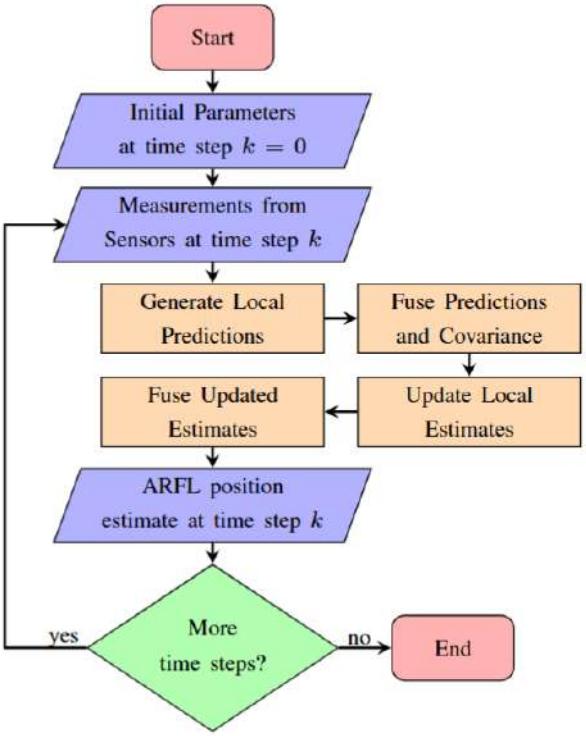


Figura 18: Fluxograma do ARFL.

a trajetória 2D de uma etiqueta BLE, cujas posições são $\begin{bmatrix} x_{\text{tag}} & y_{\text{tag}} \end{bmatrix}^T$, sob um movimento de velocidade constante ao longo de um total de K passos de teste, utilizando medições de RSSI e AoA das âncoras, processando os pacotes de dados enviados pela etiqueta BLE.

A precisão das medições de RSSI e AoA é frequentemente comprometida por ruídos, devido a fatores como efeitos de multipercurso, reflexões, refração do sinal e erros intrínsecos de leitura nos dispositivos. Além disso, a perda de pacotes devido à interferência de outras redes de RF pode reduzir ainda mais a confiabilidade. Uma alternativa para mitigar esses efeitos é o uso de filtros estocásticos, que melhoram a precisão das medições. A filtragem estocástica é uma ferramenta matemática usada para estimar o estado de um sistema ao longo do tempo na presença de incertezas. Dentre os filtros estocásticos amplamente utilizados na literatura, adota-se neste trabalho o KF [40], em conjunto com técnicas de fusão de sensores [42, 43]. A Figura 18 ilustra o fluxograma do esquema ARFL proposto.

Uma das técnicas de fusão mais conhecidas é a T2TF [42], onde as estimativas individuais de estado de diferentes sensores são combinadas para gerar uma nova estimativa do vetor de estado. Uma derivação da T2TF é a *Track Fusion Model with Fused Prediction* (TFP) [43]. Denota-se como $\hat{\mathbf{x}}_{k|k}^{f}$ a estimativa a posteriori do estado obtida pelo KF a partir do f-ésimo sensor ou técnica, com $f \in \{1, 2, \ldots, F\}$, sendo F o número total de sensores/técnicas fundidas. A TFP é adotada neste trabalho para combinar as estimativas dos algoritmos "AoA+RSSI" e "AoA-apenas". A fusão proposta é denominada ARFL.

Diferentemente da fusão em um único passo da T2TF, a TFP realiza dois passos de fusão: um para a predição fundida e outro para a estimativa de estado fundida, como ilustrado no fluxograma da Figura 18.

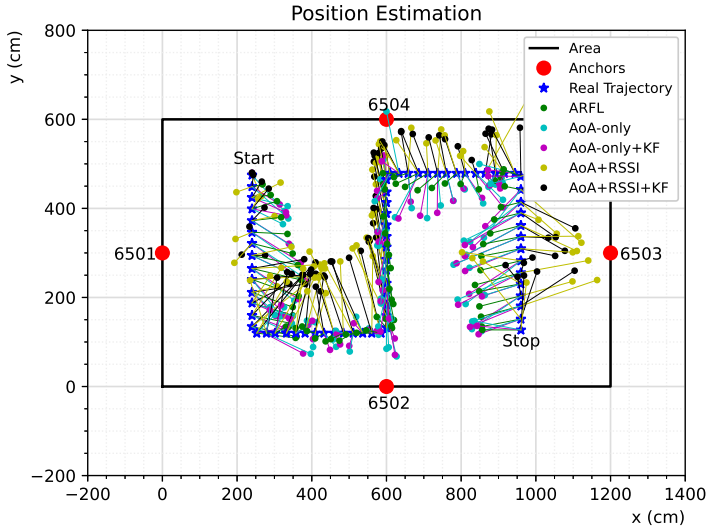


Figura 19: Estimativas de posição no Caso I. As linhas representam o erro entre a previsão e a posição real.

6.3 Resultados Simulados

O desempenho do esquema proposto é avaliado com o auxílio do conjunto de dados fornecido por [41], obtido utilizando um kit Bluetooth 5.1 u-blox XPLR-AoA-1. O ambiente experimental consiste em um espaço interno de 14m × 8m. Uma âncora foi posicionada em cada canto da sala, a uma altura de 2.3m. Cada âncora contém as seguintes medições no conjunto de dados: RSSI, ângulo de azimute, ângulo de elevação, coordenadas reais do alvo, além do tempo inicial e final.

Os cenários apresentados em [41] capturam variações dinâmicas nos dados de RSSI e AoA. Os dados foram coletados enquanto uma pessoa se movia pela sala a uma velocidade de aproximadamente 0.5 m/s. A seguir, são apresentados os resultados para um dos cenários de [41], que denominamos de Caso I. Este caso refere-se ao trajeto ilustrado pelos pontos azuis na Figura 19. Esta figura também apresenta o caminho previsto pelos métodos baseados apenas em AoA e em AoA+RSSI, com e sem KF, além da técnica ARFL. Como pode ser observado, a precisão é geralmente maior na região central da sala. Isso ocorre devido à menor dispersão angular nas âncoras, o que reduz os erros na estimativa do ângulo de azimute e, consequentemente, melhora a precisão dos métodos baseados em AoA. Além disso, a estimativa da trajetória também foi realizada utilizando multilateração. Os erros entre as posições reais e estimadas são apresentados na Figura 20, onde "MLT" representa a multilateração. O erro d_ϵ representa a distância média entre as coordenadas reais e estimadas em quatro repetições, dado por $d_\epsilon = \frac{1}{m}\sum_{i=1}^{m}\sqrt{(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2}$. Aqui, x_i e y_i são as coordenadas reais na posição i-ésima, $\hat{x}_i$ e $\hat{y}_i$ representam as coordenadas estimadas na posição i-ésima, e m denota o número total de posições estimadas ao longo do teste.

A Figura 20 revela que, entre os esquemas sem filtragem, o método baseado apenas em AoA é o mais preciso. Essa superioridade decorre de dois fatores principais. Primeiro, a triangulação baseia-se exclusivamente no AoA, eliminando a necessidade de medições de RSSI e do coeficiente de perda de percurso. Isso reduz os erros introduzidos pelos efeitos de multi-

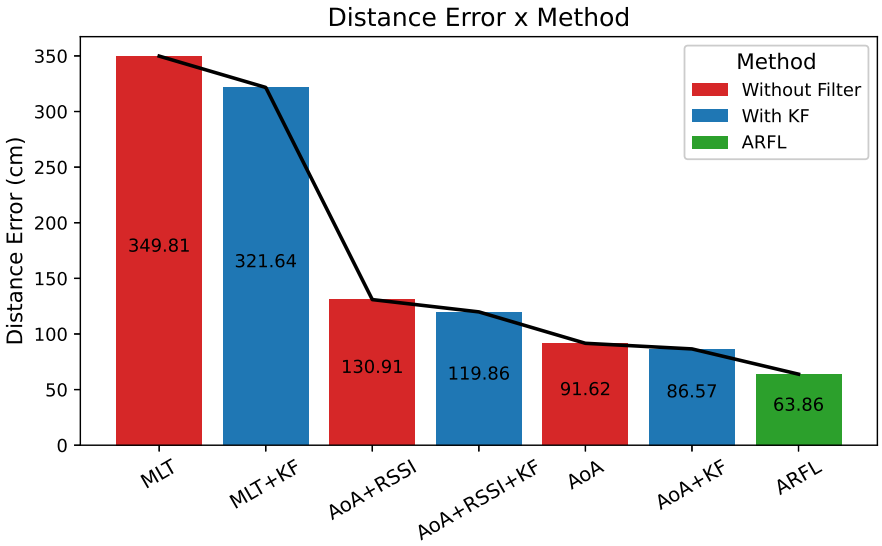


Figura 20: Gráfico de barras com os erros de distância d_ϵ para o Caso I.

percurso nessas medições. Segundo, o ângulo de azimute medido pelas âncoras se mostra uma métrica mais confiável para a estimativa de posição do que as medições de RSSI. Além disso, os resultados indicam melhorias significativas na precisão da distância ao combinar AoA com RSSI em comparação com o uso exclusivo de RSSI na multilateração. O método AoA+RSSI oferece uma melhoria de 62.57% na precisão. Em particular, a localização baseada apenas em AoA supera tanto a multilateração quanto o método AoA+RSSI por uma margem substancial, alcançando melhorias de 73.80% e 30.01% na precisão, respectivamente. Esses achados sugerem fortemente que a localização baseada em AoA oferece maior precisão para tarefas de localização em ambientes internos.

A influência do KF sobre as posições estimadas também é evidente nos resultados, representando uma melhoria média de 7.33% em relação aos resultados sem filtragem. O método proposto ARFL aprimora ainda mais o desempenho, resultando em um erro de distância de 63.86 cm, representando um ganho de precisão comparado a outros métodos com KF aplicado: 80.14% melhor que a multilateração, 46.72% melhor que AoA+RSSI e 26.33% melhor que a localização baseada apenas em AoA.

Resultados adicionais foram obtidos para outros cenários e condições, mas são omitidos aqui por questões de concisão. Esses resultados confirmaram a superioridade do método ARFL proposto.

6.4 Conclusão

Este estudo examinou algoritmos para localização em ambientes internos que aproveitam os dados obtidos por meio da tecnologia BLE. A fusão de sensores com AoA e AoA+RSSI proporcionou resultados melhores do que os algoritmos individuais, demonstrando sua eficácia. O estudo também explorou as incertezas nas medições de RSSI e seu impacto nas medições de distância, afetando os resultados da multilateração e de AoA+RSSI. Trabalhos futuros devem explorar a localização espacial em 3D, algoritmos de filtragem adicionais, novos métodos de fusão de sensores e técnicas de aprendizado de máquina para aumentar a precisão e a robustez.

7 Autoencoders Baseados em CNN para Comunicações IoT: Análise de Desempenho sob Desvanecimento $\kappa - \mu$

7.1 Introdução

Os sistemas de comunicação sem fio evoluem continuamente para atender às demandas crescentes por confiabilidade, eficiência e adaptabilidade em ambientes diversos e desafiadores. Arquiteturas tradicionais de comunicação são baseadas em blocos de processamento de sinal bem definidos, como modulação, codificação de canal e equalização, projetados e otimizados de forma independente [44, 45]. No entanto, esses sistemas convencionais frequentemente apresentam desempenho subótimo em condições complexas de desvanecimento. Aplicações em dispositivos IoT, que operam em ambientes dinâmicos com restrições energéticas e de largura de banda, exigem especialmente soluções adaptativas, tornando crucial a busca por alternativas eficientes.

Arquiteturas baseadas em *autoencoders* têm emergido como uma alternativa aos sistemas tradicionais, ao explorarem técnicas de *Deep Learning* (DL) para otimizar, de forma conjunta, os blocos de transmissão e recepção, modelando todo o sistema de comunicação como um problema de otimização *end-to-end*. O trabalho em [46] introduziu sistemas de comunicação modelados como *autoencoders*, demonstrando desempenho competitivo em BLER comparado a métodos clássicos. Essa abordagem é particularmente promissora para IoT, onde a eficiência energética e a capacidade de adaptação a cenários heterogêneos são críticas. Extensões recentes exploraram redes adversariais para comunicações multiusuário e CNNs para classificação de modulação, com desempenho comparável a métodos convencionais [46].

O DL também tem sido aplicado a desafios em comunicações sem fios, como estimação de canal e mitigação de interferências. Em [47], um *autoencoder* para mitigação de ruído com mecanismo de atenção foi proposto para prever condições de canal em comunicações milimétricas assistidas por superfícies reconfiguráveis, mostrando melhorias em qualidade de sinal e eficiência energética. Tais avanços são essenciais para redes IoT densas, onde dispositivos operam com recursos limitados. Similarmente, [48] investigou um *autoencoder* para comunicações em Terahertz, mitigando imperfeições de *hardware* e distorções de canal.

Estudos recentes também focam na robustez de *autoencoders* em condições práticas. Em [49], uma técnica de estimação de canal baseada em DL foi proposta para comunicações caóticas, usando um *autoencoder* para resistência a ruídos. Já [50] otimizou constelações para comunicação óptica sem fio, minimizando efeitos de ruído. Esses desenvolvimentos são relevantes para IoT em ambientes industriais ou urbanos, sujeitos a desvanecimento severo e interferências.

Apesar dos avanços recentes, permanece uma lacuna na literatura quanto à análise abrangente do desempenho de *autoencoders* baseados em CNNs em cenários com desvanecimento generalizado, condição frequente em implantações heterogêneas de redes IoT. Estudos anteriores limitaram-se a modelos específicos ou não avaliaram ambientes altamente dinâmicos. Esta pesquisa aborda essa lacuna ao investigar o desempenho de um *autoencoder* baseado em CNN sob o modelo de desvanecimento κ-μ, representativo de condições reais. As contribuições são:

- Análise de BLER do *autoencoder* em cenários κ-μ;
- Comparação com esquemas de modulação clássicos;

O conteúdo deste capítulo foi desenvolvido pelos pesquisadores *Pedro M. R. Pereira, Felipe A. P. de Figueiredo e Rausley A. A. de Souza.*

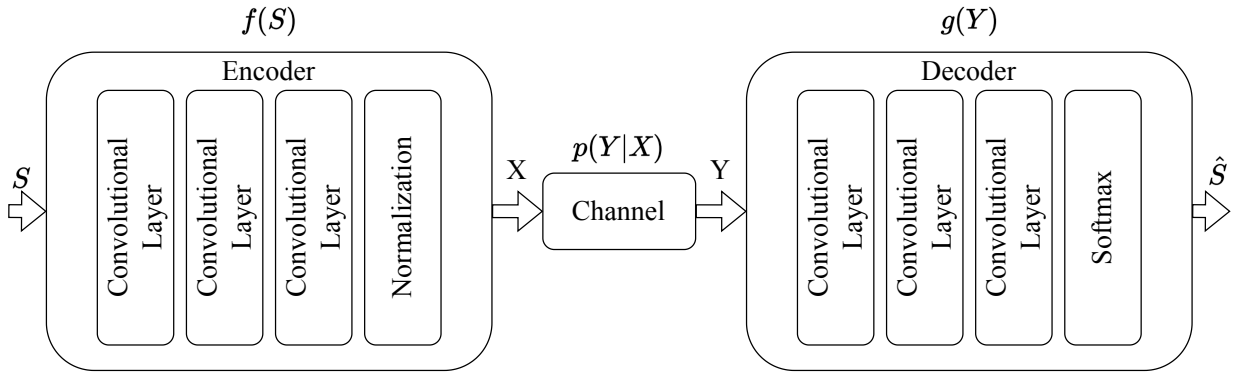


Figura 21: Sistema de comunicação *end-to-end* como um *autoencoder*.

- Resultados que demonstram adaptação eficiente a diferentes ambientes, viabilizando aplicações IoT robustas.

7.2 Modelo do Sistema

Um sistema convencional de comunicação ponto a ponto compreende três componentes principais: o transmissor, o canal e o receptor. O transmissor envia uma mensagem m pelo canal ao receptor, onde a mensagem consiste em uma sequência de L símbolos (comprimento do bloco), cada um codificando k bits de informação. Consequentemente, o número de utilizações discretas do canal é denotado como n e a taxa de transmissão do sistema é definida como $R = k/n$ (bits por uso do canal). O transmissor aplica uma função de transformação $x = f(m) \in \mathbb{C}^n$, gerando o sinal transmitido x.

O *hardware* do transmissor impõe restrições a x, representadas neste estudo como uma restrição de potência $||x||^2 \leq n$. O canal é modelado como um sistema estocástico, onde o sinal recebido segue uma *Probability Density Function* (PDF) condicional $y \sim p(y|x)$, com $y \in \mathbb{C}^n$ representando o sinal recebido. O receptor aplica então uma transformação $\hat{m} = g(y)$ para estimar a mensagem original m com erro mínimo. No contexto de DL, o transmissor e o receptor são denominados codificador e decodificador, respectivamente, e são implementados usando redes neurais, conforme ilustrado na Figura 21.

Uma rede neural *feedforward* estabelece um mapeamento $f(x_0; W) : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ de um vetor de entrada $x_0 \in \mathbb{R}^{N_0}$ para uma saída $x_L \in \mathbb{R}^{N_L}$ através de L camadas de processamento. Esse mapeamento depende dos parâmetros (ou pesos) $W = \{W_1, W_2, \ldots, W_L\}$ e da saída da camada anterior.

Este trabalho considera camadas convolucionais, onde cada camada consiste em F filtros com pesos $Q^f \in \mathbb{R}^{a \times b}$ para $f = 1, \ldots, F$, gerando um mapa de características $Y^f \in \mathbb{R}^{n' \times m'}$ a partir de uma matriz de entrada $X \in \mathbb{R}^{n \times m}$ com base na operação de convolução dada por

$$Y^f_{(i,j)} = \sum_{k=0}^{a-1} \sum_{l=1}^{b-1} Q^f_{(a-k,b-l)} X_{(1+s(i-1)-k,1+s(j-1)-l)}, \tag{8}$$

em que $s \geq 1$ representa o parâmetro de *stride* e as dimensões de saída são dadas por $n' = 1 + \left\lfloor \frac{n+a-2}{s} \right\rfloor$ e $m' = 1 + \left\lfloor \frac{m+b-2}{s} \right\rfloor$.

Tabela 3: Configuração do Autoencoder para Comunicação IoT

Bloco	**Camada**	**Ativação**	**Dimensão de Saída**
Codificador	Entrada		$L \times 2^k$
	Conv1D	ELU	$L \times 256$
	Conv1D	ELU	$L \times 256$
	Conv1D	LINEAR	$L \times 2n$
	Normalização		$L \times 2n$
Canal	Desvanecimento + Ruído		$L \times 2n$
Decodificador	Conv1D	ELU	$L \times 256$
	Conv1D	ELU	$L \times 256$
	Conv1D	SOFTMAX	$L \times 2^k$

Um *autoencoder* baseado em CNN é empregado, conforme ilustrado na Figura 21. A camada convolucional, governada por (8), permite ao transmissor processar sequências de símbolos S, manipulando $k \times L$ bits em paralelo. Cada símbolo em S é codificado como um vetor *one-hot* $O_s \in \mathbb{R}^{2^k}$, onde um único elemento é definido como um e os demais permanecem zero.

Para facilitar a codificação linear e não linear do bloco de entrada, funções de ativação *Exponential Linear Unit* (ELU) são utilizadas nas camadas convolucionais, transformando a sequência de entrada *one-hot* S em uma nova representação $X = f(S)$ através de n intervalos de canal. Isso resulta em pontos de constelação mapeados em um espaço $2n$-dimensional. Cada camada convolucional é seguida por normalização por lotes (*batch normalization*) para melhorar a estabilidade da rede, empregando 256 filtros para otimizar a representação dos símbolos de entrada.

Camadas de normalização impõem restrições de potência do transmissor mapeando representações simbólicas em um espaço $2n$-dimensional, considerando que cada um dos n intervalos de canal consiste em componentes em fase e quadratura (I/Q). A camada do canal segue a PDF condicional $p(Y|X)$ baseada na distribuição κ-μ. Além disso, *Additive White Gaussian Noise* (AWGN) com variância $\sigma^2 = \left(2RE_b/N_0\right)^{-1}$ é introduzido, onde $R = k/n$ representa a taxa de transmissão e E_b/N_0 denota a *Signal-to-Noise Ratio* (SNR).

O receptor adota uma arquitetura semelhante ao transmissor, mas omite a camada de normalização. Ele reconstrói o sinal recebido Y para classificar cada sinal entre 2^k candidatos possíveis. Assume-se que a CSI perfeita está disponível e integrada à rede do receptor juntamente com Y. Uma abordagem de decisão suave (*soft decision*) utiliza a função de ativação *softmax*, que produz um vetor de probabilidade sobre todas as sequências de entrada possíveis S. A transformação final é denotada como $\hat{S} = g(Y)$, correspondendo ao índice do elemento de maior probabilidade no vetor de saída. A Tabela 3 resume a configuração do *autoencoder* utilizado neste trabalho.

O objetivo deste sistema de auto-aprendizado é determinar o conjunto ótimo de parâmetros W^* que minimiza a função de perda $J(W)$, dada por

$$W^* = \arg\min_W J(W). \tag{9}$$

A otimização é realizada usando *Stochastic Gradient Descent* (SGD), inicializada com pesos aleatórios $W = W_0$, e atualizada iterativamente conforme

$$W_{t+1} = W_t - \eta \nabla_w \tilde{J}(W_t), \tag{10}$$

onde $\eta > 0$ é a taxa de aprendizado e ∇_w representa o gradiente da perda *Binary Cross Entropy* (BCE) aproximada $\tilde{J}(W)$, calculada para um *minibatch* $N_t \subset \{1, 2, \ldots, N\}$ de tamanho N_t em cada iteração, definida por

$$\tilde{J}(W) = -\frac{1}{N_t} \sum_{i \in N_t} S_i \log(g(Y_i)). \tag{11}$$

A perda é otimizada via retropropagação (*backpropagation*) sobre um conjunto de dados de tamanho N, facilitando a melhoria de desempenho do sistema de comunicação baseado em *autoencoder*.

7.2.1 Modelo de Canal

O modelo de desvanecimento κ-μ representa uma distribuição estatística generalizada que caracteriza variações em pequena escala em sinais de desvanecimento sob condições de *Line of Sight* (LOS) e *Non-Line of Sight* (NLOS). Diferentemente de modelos convencionais, a distribuição κ-μ considera *clusters* de ondas de multipercurso e a não linearidade do ambiente de propagação. Modelos conhecidos, como Exponencial, Rayleigh, Nakagami-m, Gama e Weibull, são casos especiais da distribuição κ-μ. O parâmetro κ descreve a razão de potência entre componentes dominantes e ondas espalhadas, enquanto μ representa o número de *clusters* de multipercurso. Consequentemente, a envoltória do sinal segue uma função não linear baseada na soma de componentes multipercurso, sendo expressa como

$$r^2 = \sum_{i=1}^{n} (x_i + p_i)^2 + \sum_{i=1}^{n} (y_i + q_i)^2, \tag{12}$$

onde x_i e y_i são processos Gaussianos independentes com expectativas $E(x_i) = E(y_i) = 0$ e variâncias $E(x_i^2) = E(y_i^2) = \sigma^2$. Além disso, p_i e q_i denotam os valores médios dos componentes em fase e quadratura dos *clusters* multipercurso.

Dado um sinal de desvanecimento com envoltória r e envoltória normalizada $\rho = r/\hat{r}$, onde $\hat{r} = \sqrt{E(r^2)}$ representa o valor *Root Mean Square* (RMS) de r, a PDF de ρ é definida como [51, eqn.(1)]

$$p(\rho) = \frac{2\mu(1+\kappa)^{\frac{\mu+1}{2}}}{\kappa^{\frac{\mu-1}{2}} e^{\mu\kappa}} \rho^{\mu} e^{-\mu(1+\kappa)\rho^2} I_{\mu-1}\left(2\mu\sqrt{\kappa(1+\kappa)}\rho\right), \tag{13}$$

onde $\kappa \geq 0$ é a razão entre a potência total dos componentes dominantes e a das ondas espalhadas, e $\mu \geq 0$ é dado por

$$\mu = \frac{E^2(r^2)}{\mathrm{Var}(r^2)} \times \frac{1 + 2\kappa(1+\kappa)^2}{(1+\kappa)^2}. \tag{14}$$

Além disso, a restrição $\frac{\mu(1+\kappa)^2}{1+2\kappa} \geq \frac{1}{2}$ deve ser satisfeita. Aqui, $I_v(.)$ denota a função de Bessel modificada de primeira espécie e ordem v.

Vários modelos de desvanecimento conhecidos podem ser derivados como casos especiais da distribuição κ-μ. A distribuição de Weibull é obtida definindo $\mu = 1$ e ajustando κ adequadamente. A distribuição de Rayleigh surge quando $\mu = 1$ e $\kappa = 0$. A distribuição Nakagami-m é um caso específico do modelo κ-μ onde $\kappa = 0$, com m representando os *clusters* de multipercurso.

7.2.2 BLER em Canais com Desvanecimento $\kappa - \mu$

Para derivar o BLER em canais com desvanecimento plano κ-μ, definimos a SNR instantânea como $\gamma = R^2$, onde R representa a envoltória de desvanecimento e a SNR média é dada por $\bar{\gamma} = \mathbb{E}[\gamma]$.

Aplicando técnicas padrão de transformação de variáveis aleatórias a partir de ρ, a PDF de γ é expressa como

$$f_\gamma(\gamma) = \frac{2\mu(1+\kappa)^{(\mu+1)/2}}{\Gamma(\kappa^{(\mu-1)/2}\exp(\mu\kappa))}\gamma^{\mu/2-1} \times e^{-\mu(1+\kappa)\gamma} I_{\mu-1}(2\mu\sqrt{\kappa(1+\kappa)\gamma}). \tag{15}$$

A probabilidade de erro incondicional P_e é definida como

$$P_\mathrm{e} = \int_0^\infty p_\mathrm{e}(\gamma) f_\gamma(\gamma) d\gamma, \tag{16}$$

onde $p_\mathrm{e}(\gamma)$ depende do esquema de modulação. Por exemplo, em esquemas de *M-Quadrature Amplitude Modulation* (MQAM), a probabilidade de erro é aproximada por

$$p_\mathrm{e}(\gamma) = Q\left(\sqrt{\frac{3\gamma \log_2 M}{M-1}}\right), \tag{17}$$

onde $Q(.)$ representa a função Q, definida sendo $Q(x) = \frac{1}{2\pi}\int_x^\infty \exp(-u^2/2)du$, e M é a ordem de modulação. Para sistemas *Binary Phase Shift Keying* (BPSK) e *Quadrature Phase Shift Keying* (QPSK), a probabilidade de erro pode ser aproximada para

$$p_\mathrm{e}(\gamma) = Q(\sqrt{2\gamma}). \tag{18}$$

7.3 Trabalhos Futuros

A pesquisa está em progresso, mas de acordo com o cronograma de atividades. Até novembro de 2025 haverão resultados simulados para avaliar o desempenho da proposta. Os próximos passos são listado abaixo:

1. Implementação do *Autoencoder*
   - Otimização da arquitetura CNN (número de camadas, filtros, funções de ativação);
   - Integração do modelo de canal κ-μ no *pipeline* de simulação;
   - Validação da restrição de potência ($||\mathbf{x}||^2 \leq n$) e normalização no codificador.
2. Simulação em Cenários Diversificados
   - Variar parâmetros κ e μ;
   - Testar valores de E_b/N_0 (i.e., SNR) para avaliar robustez em baixa potência;
   - Comparar BLER do *autoencoder* com esquemas clássicos sob as mesmas condições.
3. Análise de Adaptabilidade

- Avaliar ajustes a mudanças dinâmicas no canal (e.g., transições LOS/NLOS);
- Incorporar cenários com mobilidade (variação temporal de κ-μ).

4. Otimização de Hiperparâmetros
   - Ajustar a taxa de aprendizado (η), tamanho do *batch* e regularização para minimizar perda BCE;
   - Explorar alternativas ao otimizador SGD (e.g., Adam, RMSProp) para convergência acelerada.
5. Publicação de Resultados
   - Consolidar dados de BLER em gráficos comparativos;
   - Escrever uma análise crítica sobre vantagens/limitações da abordagem proposta;
   - Submissão do trabalho.

8 Detecção Inteligente de Incêndios Florestais Usando Modelos de Aprendizado Profundo

8.1 Introdução

Incêndios florestais caracterizam-se pela combustão rápida e descontrolada de biomassa vegetal, incluindo folhas, galhos e árvores, resultando na degradação extensiva de áreas naturais e na disrupção de ecossistemas. Esses eventos representam uma ameaça crítica à biodiversidade, promovem alterações microclimáticas e atmosféricas e comprometem a saúde pública devido à emissão de poluentes atmosféricos, como monóxido de carbono, material particulado e compostos orgânicos voláteis. As causas podem ser de origem natural, como descargas atmosféricas, ou antrópicas, como práticas agropecuárias irregulares, queimadas intencionais e expansão desordenada da fronteira agrícola. A detecção precoce é estratégica para a mitigação de danos ambientais e humanos, demandando sistemas integrados de monitoramento e resposta, com a participação coordenada de órgãos governamentais, comunidades locais e tecnologias emergentes [52, 53, 54].

No Brasil, a incidência de incêndios florestais tem apresentado uma tendência crescente e preocupante. Entre os anos de 1985 e 2023, estima-se que mais de 10 milhões de hectares de vegetação nativa foram consumidos por queimadas. O bioma Caatinga, exclusivamente brasileiro e caracterizado por uma vegetação xerofítica, apresenta, em média, 0,56% de sua extensão comprometida anualmente por incêndios. Em 2022, foram registrados aproximadamente 85 mil focos ativos de incêndio no território nacional. Esse número aumentou para 102 mil em 2023, e em 2024 o país alcançou um novo recorde, com mais de 110 mil focos registrados, correspondendo à devastação de uma área estimada em 114 mil km^2, um crescimento de 116% em relação ao ano anterior. A intensificação desses eventos, fortemente associada a períodos prolongados de estiagem e à ocorrência do fenômeno climático El Niño, reforça a necessidade urgente de desenvolver e implementar soluções tecnológicas robustas para o monitoramento, a prevenção e o combate a incêndios florestais [55, 56].

Considerando o agravamento do cenário atual, a detecção precoce de incêndios florestais torna-se um elemento crítico para a mitigação dos impactos ambientais, sociais e econômicos. A identificação de focos de calor ainda em estágios iniciais viabiliza ações de resposta mais rápidas e eficazes, contribuindo para a contenção da propagação do fogo e a preservação de ecossistemas, bem como para a redução dos riscos à saúde pública. As estratégias tradicionais de monitoramento incluem torres de vigilância, estruturas posicionadas em pontos estratégicos de áreas florestais, com alturas variando entre 15 e 50 metros, dependendo das características topográficas, além de patrulhamento aéreo e análise de séries temporais de dados históricos. No entanto, essas abordagens convencionais apresentam limitações quanto à cobertura espacial e à acurácia em tempo real, fatores essenciais para decisões operacionais em contextos de emergência. Frente às crescentes ameaças provocadas pelas mudanças climáticas e à complexidade intrínseca dos ecossistemas florestais, torna-se imperativo adotar soluções baseadas em tecnologias emergentes, capazes de fornecer monitoramento contínuo, detecção automatizada e resposta integrada.

Neste trabalho, propõe-se uma abordagem avançada para a detecção e análise de incêndios florestais com base em algoritmos de aprendizado profundo. O principal objetivo é monitorar

---

O conteúdo desta seção consiste em um resumo adaptado pelo pesquisador *Samuel Baraldi Mafra*, com base no artigo de Tarciso G. B. de Bello e Samuel B. Mafra, "Detecção Inteligente de Incêndios Florestais Usando Modelos de Aprendizado Profundo," submetido ao SBrT 2025.

a dinâmica dos focos de incêndio, identificando padrões de variação temporal que indiquem se o fogo está em fase de expansão ou contenção. Essa análise permite não apenas a detecção precoce, mas também o acompanhamento contínuo da evolução do incêndio, subsidiando ações mais precisas de resposta e alocação de recursos.

8.2 Solução Proposta

Neste trabalho, propõe-se uma solução baseada em IoT para o monitoramento em tempo real e a detecção de focos de incêndio, utilizando algoritmos de aprendizado profundo. Foram selecionados os modelos R-CNN, YOLOv8 e YOLOv11, considerando seu elevado desempenho, ampla adoção em estudos correlatos e capacidades computacionais. A metodologia adotada consiste em uma análise comparativa entre esses modelos sob condições experimentais similares, com o objetivo de avaliar suas respectivas métricas de desempenho, como acurácia, precisão, revocação e tempo de inferência. O modelo R-CNN foi incluído por sua abordagem baseada em múltiplas passagens e capacidade de análise detalhada de regiões específicas da imagem. Por outro lado, os modelos YOLOv8 e YOLOv11 foram escolhidos por sua arquitetura de detecção de passagem única (*single-shot*), que permite processamento mais rápido e eficiente, tornando-os particularmente adequados para aplicações em tempo real.

Neste contexto, são empregados diferentes modelos de aprendizado profundo com o objetivo de realizar a detecção precisa de focos de incêndio em ambientes florestais. Essa tarefa apresenta elevada complexidade devido à variabilidade temporal e espacial das condições ambientais durante os incêndios, como alterações climáticas, presença de fumaça, variações de luminosidade e heterogeneidade da cobertura vegetal. Os modelos analisados foram aplicados tanto à identificação de focos de pequena quanto de grande extensão, abrangendo diferentes tipos de vegetação e cenários ambientais. A Figura 22 exemplifica uma dessas situações, onde é possível identificar focos de incêndio em uma área de vegetação.

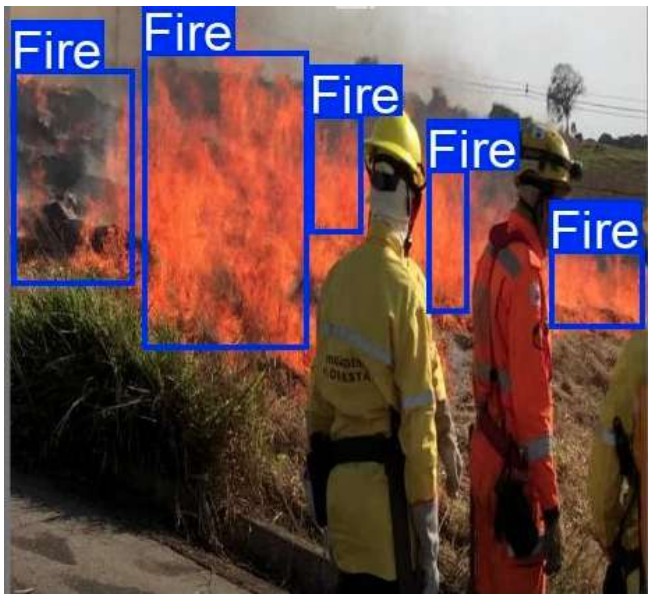


Figura 22: Imagem do Treinamento - Detecção de Incêndios

Com o objetivo de detectar o aumento ou a redução de focos de incêndio, propõe-se o sistema ilustrado na Figura 23, no qual uma câmera é responsável pela captura das imagens dos focos, enquanto um dispositivo computacional, como a Raspberry Pi, pode ser utilizado

para o processamento das imagens e o envio dos dados coletados, como em um cenário típico de IoT.

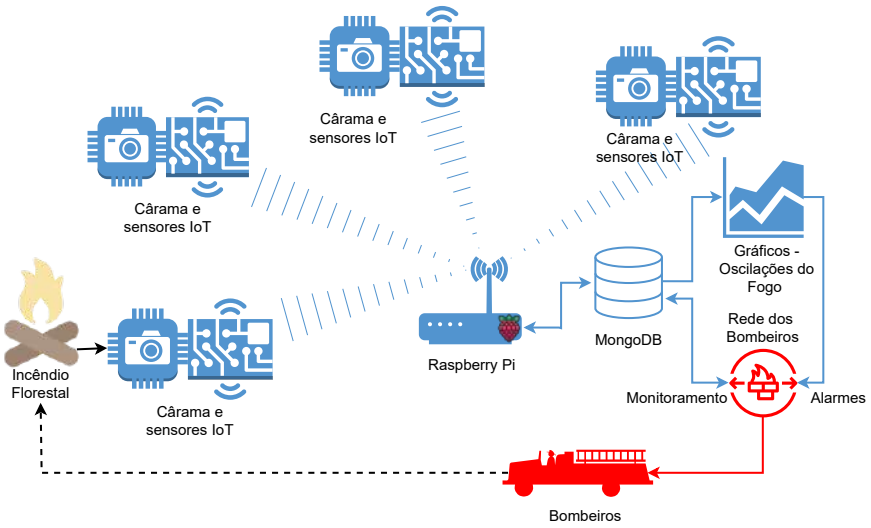


Figura 23: Modelo do Sistema IoT Proposto

A coleta e o treinamento de dados constituem etapas fundamentais para assegurar que os modelos de detecção sejam expostos a um conjunto de treinamento diversificado e representativo das condições reais associadas a incêndios florestais. A construção do banco de imagens, ou *dataset*, foi orientada pela necessidade de contemplar tanto cenários com presença de fogo quanto situações de normalidade, abrangendo diferentes condições ambientais, tipos de vegetação e variações de iluminação. As imagens utilizadas foram obtidas durante treinamentos operacionais realizados pelo Corpo de Bombeiros de Pouso Alegre, Minas Gerais, Brasil. As coletas foram realizadas por meio de drones e câmeras posicionadas em múltiplos pontos da área queimada, permitindo a obtenção de um conjunto de dados heterogêneo e adequado ao treinamento robusto dos modelos propostos.

Após a etapa de seleção das imagens, procede-se à fase de rotulagem, a qual desempenha um papel fundamental na qualidade do treinamento dos modelos de detecção, como R-CNN, YOLOv8 e YOLOv11. A acurácia do processo de anotação impacta diretamente a capacidade dos modelos em aprender a identificar, de forma precisa, os objetos de interesse — neste caso, os focos de incêndio. A rotulagem foi realizada manualmente, por meio da inserção de caixas delimitadoras (*bounding boxes*) sobre as regiões afetadas pelo fogo, permitindo a marcação exata das áreas relevantes nas imagens. Para essa tarefa, utilizou-se a plataforma Roboflow, que oferece um ambiente integrado para anotação, gerenciamento e exportação de *datasets*, otimizando o fluxo de trabalho e garantindo maior consistência nos dados utilizados para treinamento e validação dos modelos.

Após a etapa de rotulagem, as 80 imagens selecionadas foram particionadas em três conjuntos: 70% destinadas ao treinamento, 15% à validação e 15% ao teste. Essa divisão visa maximizar o aprendizado do modelo a partir do maior volume possível de dados, ao mesmo tempo em que permite monitorar o desempenho durante o processo de treinamento (validação) e avaliar sua capacidade de generalização em dados não vistos (teste). Em seguida, os conjuntos de imagens foram integrados aos modelos R-CNN, YOLOv8 e YOLOv11 para a fase de treinamento supervisionado.

8.3 Resultados Experimentais

Com o objetivo de avaliar a eficácia da metodologia proposta, os experimentos foram conduzidos utilizando o banco de dados de imagens de incêndios florestais descrito na seção anterior. Para os testes, todos os modelos de aprendizado profundo, R-CNN, YOLOv8 e YOLOv11, foram treinados por 2000 épocas, com imagens redimensionadas para 640×640 pixels e tamanho de lote (*batch size*) igual a 16. Os resultados obtidos indicam que o modelo YOLOv11 apresentou desempenho superior em comparação aos demais, conforme evidenciado na Tabela 4. A evolução do treinamento do YOLOv11 é ilustrada na Figura 24, a qual mostra a redução progressiva das funções de perda (*loss functions*) ao longo das iterações, tanto para o conjunto de treinamento quanto para o de validação, sinalizando uma convergência estável do modelo. Tal comportamento é indicativo de um processo de aprendizado eficiente, com adequada adaptação aos dados e ausência de indícios de *overfitting*. A trajetória semelhante observada na curva de perdas do conjunto de validação reforça a capacidade de generalização do modelo frente a dados não previamente vistos.

Tabela 4: Comparação do Desempenho dos Modelos de Detecção de Incêndios Florestais.

Modelos	Acurácia	Precisão	Recall	F1-Score
Yolov11	98,60%	98,97%	98,90%	98,97%
Yolov8	98,10%	98,10%	98,30%	98,35%
RCNN	94,65%	95,80%	96,85%	96,87%

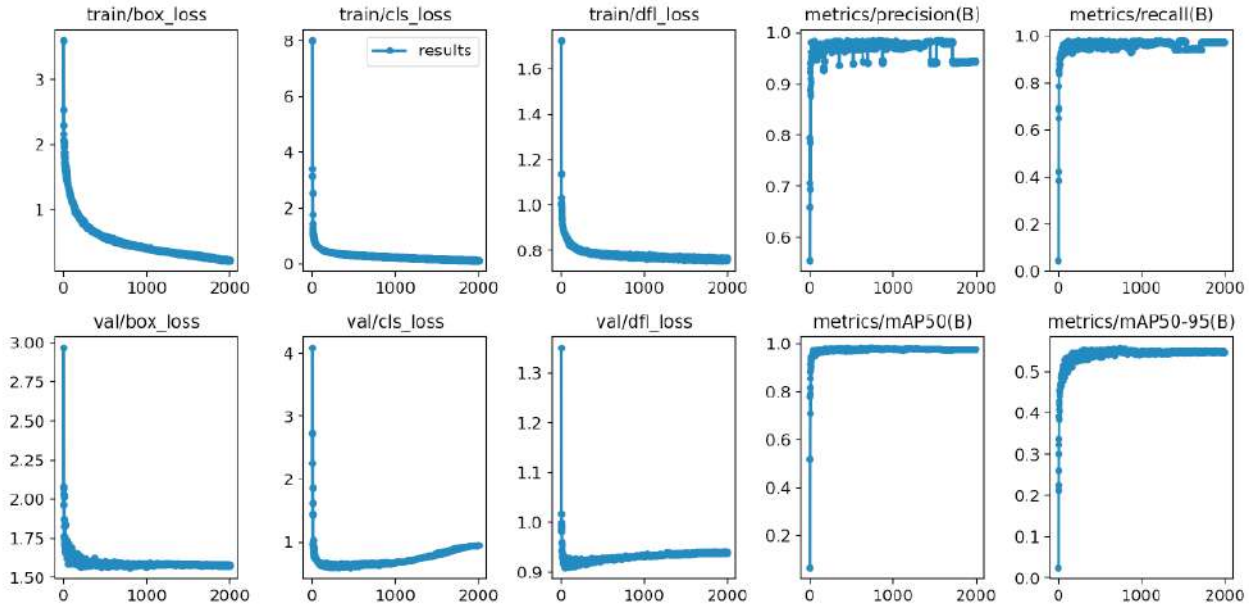


Figura 24: Evolução do treinamento do YOLOv11.

As subfiguras correspondentes às métricas de *Precision*, *Recall* e *Mean Avarage Precision* (mAP) fornecem uma análise detalhada da evolução do desempenho do sistema ao longo das épocas de treinamento. Essas métricas são fundamentais para avaliar a capacidade dos modelos em detectar focos de incêndio com precisão, mantendo um equilíbrio entre sensibilidade e especificidade, sem comprometer a eficiência em tempo real. As variações observadas nas

curvas de validação podem ser interpretadas como reflexos da adaptação do modelo às variações nos dados, enquanto a estabilidade das métricas durante as iterações indica a robustez e a consistência do YOLOv11 frente ao processo de generalização. O acompanhamento contínuo desses indicadores ao longo das fases de treinamento e validação é essencial para garantir que o modelo alcance o melhor desempenho possível na tarefa de detecção de incêndios florestais.

8.4 Conclusão

A proposta apresentada neste trabalho evidencia o potencial dos modelos de aprendizado profundo na tarefa de detecção automatizada de incêndios florestais. Conforme demonstrado nos experimentos, o modelo YOLOv11 destacou-se em relação ao YOLOv8 e ao R-CNN, apresentando melhor desempenho nas métricas avaliadas. Apesar dos resultados promissores, ainda persistem desafios relevantes, como a necessidade de um conjunto de dados mais abrangente e representativo, que contemple diferentes condições ambientais, tipos de vegetação e variações sazonais. Além disso, a inclusão de novas classes, como a detecção de fumaça e outros indícios precoces de incêndio, pode contribuir significativamente para o aumento da acurácia e da capacidade preditiva dos modelos.

O trabalho está em progresso. Até dezembro de 2025, a solução IoT completa será implementada e testes com os dispositivos embarcados serão realizados em campo.

9 Proteção de Serviços URLLC para Indústria 4.0 Utilizando Aprendizagem por Reforço Multi-Agente

9.1 Introdução

Nas redes 6G, o conceito de *Network Slicing* (NS) tem recebido especial interesse ao permitir o compartilhamento de recursos entre os diferentes nós da rede. Esse conceito surgiu do recente avanço nas tecnologias de computação e virtualização de funções e possibilita o fornecimento de serviços personalizados para cada cenário de aplicação distinto. O NS é comumente usado para alocação de recursos a fim de garantir o bom desempenho de serviços heterogêneos na Quinta Geração de Rede Móvel Celular (5G). Esses serviços incluem *enhanced Mobile Broadband* (eMBB), *Massive Machine Type Communication* (mMTC) e URLLC [57]. Nesses serviços, questões relacionadas a NS precisam ser analisadas, como orquestração *intra-slice* e *inter-slice*, bem como alocação dinâmica de recursos. Os principais desafios estão relacionados à largura de banda limitada e aos requisitos de *slices* heterogêneos, os quais exigem planejamento cuidadoso e técnicas de compartilhamento eficientes [58].

Uma alternativa que tem se mostrado bastante eficaz para melhorar a QoS em NS é o uso de algoritmos de aprendizagem de máquina. Em particular, o uso de agentes de *Reinforcement Learning* (RL) é capaz de escalonar os recursos de rede para atender a requisitos rigorosos de variados casos de uso das futuras redes de comunicação sem fio [58]. Entretanto, algumas abordagens na literatura que utilizam algoritmos de RL sofrem com o provisionamento excessivo e/ou violações frequentes do *Service Level Agreements* (SLA), já que não garantem o cumprimento dos requisitos ou proteção de *slices* de rede que são essenciais para o *Radio Resource Scheduling* (RRS) em NS da RAN [59]. Em [60], os autores propõem um problema de MARL considerando um cenário de rede celular densa que contém vários *slices* de rede em múltiplas BSs com o objetivo de gerenciar os recursos entre *slices* em tempo real. Em [61], os autores apresentam um novo algoritmo MARL para fatiamento de uma rede *Open* RAN, projetado para se adaptar a números de fatias variáveis e escalar efetivamente à medida que aumentam. Já em [62], é proposto um método baseado em RL hierárquico para alocar recursos de rádio para usuários móveis que utilizam serviços do tipo URLLC e eMBB. Considerando cenários da Indústria 4.0, Nahum *et al.* [63] introduzem um agente RL de RRS para NS capaz de orquestrar recursos de rádio entre vários *slices* para atender aos seus requisitos e dar prioridade à latência e à confiabilidade, resultando em menos violações de *slice* com foco na proteção de aplicações críticas (*slice* URLLC). Essa proteção no âmbito da Indústria 4.0 é uma demanda essencial para o RRS devido às variações de rede e canal do dispositivo, que podem causar instabilidade, um cenário que requer adaptações constantes do RRS para atender aos requisitos críticos da aplicação.

Neste trabalho, propõe-se um novo escalonador MARL utilizando o método SAC. A proposta estende o agente introduzido em [63] a fim de dar suporte a RL nas funções *inter-slice* e *intra-slice* do escalonador. Como resultado, observa-se um aprimoramento na proteção aos *slices* críticos em aplicações da Indústria 4.0, para os quais o RRS é priorizado, melhorando portanto os requisitos de QoS.

O conteúdo deste capítulo foi desenvolvido pelos pesquisadores *Cleverson Veloso Nahum, Weskley Vinicius Fernandes Mauricio, Maykon Renan Pereira da Silva, Francisco Raimundo Albuquerque Parente e Aldebaro Klautau.*

9.2 Modelo de Sistema

Neste trabalho, é considerado o aspecto de *downlink* de um sistema MIMO industrial *indoor* que utiliza a tecnologia *Frequency Division Duplex* (FDD) *Orthogonal Frequency Division Multiple Access* (OFDMA), como em um galpão de fábrica. Esse sistema é composto por uma BS equipada com um modelo de antena 3D $n_v \times n_h$ [64], servindo U UEs, cada um equipado com antenas únicas omnidirecionais, onde n_v e n_h são o número de antenas verticais e horizontais, respectivamente. Dentro deste contexto, são identificados três tipos de *slices* distintos, cada um adaptado para casos de uso específicos, com requisitos únicos de atraso e taxas de dados alcançáveis. Para fins de simplificação da análise, assume-se que cada UE pertence exclusivamente a um desses tipos de *slices*. Além disso, considera-se R *Resource Block*s (RBs) disponíveis para cada *Transmission Time Interval* (TTI). Um RB representa a menor unidade de recursos que pode ser alocada para uma UE. Cada RB é subdividida em N_{sc} subportadoras OFDMA adjacentes e N_{symb} símbolos *Orthogonal Frequency Division Multiplexing* (OFDM) consecutivos.

Em particular, utilizamos a técnica de precodificação em diversidade *Maximum Ratio Transmission* (MRT), que pode ser definida como

$$\mathbf{x} = \frac{\mathbf{h}^{\mathrm{H}}}{\|\mathbf{h}\|_{\mathrm{F}}}, \tag{19}$$

onde $\mathbf{x}$ é o vetor de precodificação $n_v \cdot n_h \times 1$, $\mathbf{h}$ é o vetor de canal *downlink* $1 \times n_v \cdot n_h$ entre BS e UE e $\|.\|_{\mathrm{F}}$ é a norma de Frobenius. Assim, define-se o vetor de canal efetivo do *downlink* como $\mathbf{h}_u^{\mathrm{eff}}$ do UE u após a aplicação do precodificador MRT como

$$\mathbf{h}_u^{\mathrm{eff}} = \mathbf{h}_u \mathbf{x}_u, \tag{20}$$

onde $\mathbf{h}_u$ é o vetor de canal *downlink* $1 \times n_v \cdot n_h$ do UE u. Dado que a BS aloca o r-ésimo RB para o u-ésimo UE, a SNR percebida pelo UE pode ser definida como

$$\gamma_{u,r} = \frac{\alpha_j \; p_r \; |h_{u,r}^{\mathrm{eff}}|^2}{\sigma_u^2}, \tag{21}$$

onde p_r é a potência de transmissão alocada para o r-ésimo RB, α_u é o efeito do ganho de percurso e sombreamento sofrido pelo u-ésimo UE, e σ_u é a potência do ruído sofrido pelo u-ésimo UE. Desta forma, a eficiência espectral $\mathsf{S}_{u,n}(n)$ do RB r e UE u é definida como

$$\mathsf{S}_{u,r}(n) = \log_2(1 + \gamma_{u,r}). \tag{22}$$

Além disso, emprega-se o modelo de canal *Quasi Deterministic Radio Channel Generator* (QuaDRiGa) considerando LOS e NLOS, como documentado na referência [65]. O cenário adotado foi especificamente projetado para aplicações industriais em ambientes *indoor*, como galpões fabris, refletindo as características típicas da Indústria 4.0. Para uma compreensão abrangente dos parâmetros de canal usados para criar este cenário, tais como detalhes sobre *shadow fading*, *small fading* e *angular spread*, o leitor pode consultar a referência [65].

9.2.1 *Slicing* de Rede e Requisitos do *Slice*

Cada *slice* de rede, denominado como s, engloba um grupo de UEs U_s que possuem requisitos idênticos de QoS e comportamento de tráfego semelhante. O vetor $\mathbf{R}_n$ contém o número de

Resource Block Groups (RBGs) alocados para cada *slice* pelo escalonador *inter-slice* no passo de simulação n, que pode ser definido como

$$\mathbf{R}_n = [R_1(n), R_2(n), \ldots, R_S(n)], \tag{23}$$

onde $R_s(n)$ representa o número de RBGs alocados para o *slice* s no passo n.

O princípio RRS adere a um processo dado por

$$\sum_{s=1}^{S} R_s(n) = R, \tag{24}$$

garantindo que a alocação agregada dos RBGs em todos os *slices* corresponda consistentemente ao total de RBGs disponíveis, designado como R. Consequentemente, o papel primário do RRS em um cenário que envolve a segmentação da RAN é determinar a alocação do $R_s(n)$ para cada *slice* s em um passo n específico, alinhando-a com as condições da rede para cumprir os requisitos específicos do *slice*. Depois que os RBs foram alocados entre os vários *slices*, o escalonamento *intra-slice* assume a responsabilidade de distribuir esses RBs entre os UEs individuais dentro de cada *slice*.

As exigências dos *slices* estabelecem valores desejados específicos para várias métricas de rede monitoradas. Essas exigências são formuladas com foco em duas métricas-chave: taxa de transferência atendida e atraso de *buffer*. Neste trabalho, as métricas de desempenho da rede são calculadas conforme definido em [66], considerando a taxa de transferência atendida (taxa de dados alcançada) $r_u(n)$, a taxa de ocupação do *buffer* $b_u^{\text{occ}}(n)$ e o atraso do *buffer* $d_u(n)$.

A taxa de transferência atendida para cada UE é a taxa máxima em bits por passo que um UE pode obter, considerando o número de RBGs alocado a ele e sua eficiência espectral, dada por

$$r_u(n) = \left\lfloor \frac{(R_s^u(n)/R)\text{BS}_u(n)}{P} \right\rfloor P, \tag{25}$$

onde $R_s^u(n)$ representa o número de RBGs alocados ao u-ésimo UE pelo escalonador *intra-slice*, B é a largura de banda total disponível na BS, $S_u(n)$ é a eficiência espectral para o u-ésimo UE no passo n e P é o tamanho do pacote. Já a taxa de ocupação do *buffer* é definida como

$$b_u^{\text{occ}}(n) = \frac{b_u(n)}{b_{\max}}, \tag{26}$$

onde $b_u(n)$ representa a quantidade de dados disponível no *buffer* do u-ésimo UE no passo de simulação n e $b_{\max}$ é a capacidade máxima do *buffer* do UE. Pacotes são perdidos toda vez que o *buffer* está cheio ou o atraso do pacote excede o atraso máximo $d_{\max}$ permitido pelo *buffer*. O atraso do *buffer* representa o tempo médio que cada pacote esperou antes de ser enviado ou perdido, sendo definido como

$$d_u(n) = \frac{\sum_{i=0}^{d_{\max}} i d_n^u(i)}{\sum_{i=0}^{d_{\max}} d_n^u(i)}, \tag{27}$$

onde $d_u(n)$ é um vetor de dimensão $d_{\max} + 1$ representando o atraso dos pacotes no *buffer* do u-ésimo UE no passo n.

As métricas de desempenho de cada *slice* da rede são caracterizadas por meio da média das métricas individuais calculadas para todos os UEs associados ao respectivo *slice*. Cada tipo de *slice* é associado a um conjunto específico de requisitos de desempenho, refletindo diferentes

casos de uso. Neste estudo, são considerados três tipos distintos de *slice* — eMBB, URLLC e mMTC — cujas características são descritas a seguir.

***Slice* eMBB**: UEs alocados para o *slice* eMBB priorizam altas taxas de dados, independentemente das condições do canal, e tem requisitos menos restritos relativos a atraso e perda de pacotes. Neste estudo, são definidos dois requisitos principais para o *slice* eMBB, conforme descrito a seguir.

- Taxa de transferência $r_{\text{embb}}(n)$: a taxa de transferência para o *slice* eMBB deve satisfazer ou exceder um limite mínimo especificado, denotado como $r_{\text{embb}}^{\text{req}}$.
- Atraso de *buffer* $d_{\text{embb}}(n)$: o atraso do *buffer* para o *slice* eMBB precisa ser mantido abaixo de um limite de atraso do *buffer* predeterminado, denominado como $d_{\text{embb}}^{\text{req}}$.

A taxa de transferência requisitada pelos UEs associados com o *slice* eMBB é denotada como $r_{\text{u}}^{\text{rqt}}(n)$ seguindo uma distribuição de Poisson com um valor médio de μ_{embb}.

***Slice* URLLC**: Os UEs designados para um *slice* URLLC priorizam a latência mínima e a comunicação altamente confiável, geralmente caracterizada por uma taxa de perda de pacotes extremamente baixa. Os *slices* URLLC geralmente servem aplicações para missões críticas, necessitando de priorização da rede para garantir a adesão aos seus requisitos específicos. As métricas avaliadas para URLLC são similares às usadas para o *slice* eMBB e estão descritas abaixo.

- Taxa de transferência $r_{\text{urllc}}(n)$: representa a taxa de transferência alcançada para o *slice* URLLC. Além disso, o $r_{\text{urllc}}^{\text{req}}$ denota a taxa de transferência alvo que o *slice* URLLC visa alcançar.
- Atraso do *buffer* $d_{\text{urllc}}(n)$: representa o atraso sofrido no *buffer* do *slice* URLLC. Além disso, $d_{\text{urllc}}^{\text{req}}$ denota o atraso de *buffer* desejado para o tráfego URLLC.

A taxa de transferência requisitada $r_{\text{u}}^{\text{rqt}}(n)$ dos UEs u URLLC é definida como uma distribuição de Poisson com média μ_{urllc}.

***Slice* mMTC**: O *slice* mMTC é projetado para facilitar conexões para um grande número de UEs. Para o mMTC, considera-se que o atraso do *buffer*, representado como $d_{\text{mmtc}}(n)$, deve ser mantido igual ou abaixo de um limite especificado de atraso do *buffer*, denominado $d_{\text{mmtc}}^{\text{req}}$. Além disso, no *slice* mMTC, os UEs têm um padrão de ativação probabilístico. Em cada passo de simulação, há uma probabilidade de 50% de que os UEs mMTC sejam ativados ou desativados. A taxa de transferência requisitada para o UE u, , pertencente ao *slice* mMTC $r_{\text{u}}^{\text{rqt}}(n)$, é modelada como uma variável aleatória com distribuição de Poisson de média μ_{mmtc}, no caso de o UE estar ativado. Caso contrário, essa taxa é considerada nula.

Devido às flutuações nas condições dos canais dos UEs, podem ocorrer situações em que os recursos de rádio disponíveis na rede sejam insuficientes para satisfazer os requisitos dos *slices* ao longo da simulação. Diante desse cenário, e considerando a natureza crítica das aplicações URLLC, torna-se fundamental priorizar seus requisitos acima dos requisitos dos demais tipos de aplicação.

9.3 Escalonador MARL Proposto

Nesta seção, propõe-se um escalonador MARL utilizando o método SAC para desempenhar as funções de escalonamento de *inter-slice* e *intra-slice*. O agente proposto em [63] foi estendido para dar suporte à utilização de RL nas funções do escalonador *inter-slice* e *intra-slice*, com o intuito de melhorar a proteção aos *slices* críticos em cenários industriais através da priorização da alocação de recursos para esses *slices* quando a quantidade de recursos não é suficiente para atender a todos os requisitos.

O agente de *inter-slice* é o responsável por definir a quantidade de RBs para cada um dos *slices*, enquanto os agentes de *intra-slice* são responsáveis por alocar os RBs disponibilizados pelo agente de *intra-slice* para os UEs. O agente de *inter-slice* implementa o método de RL SAC [67] tendo uma política própria e não compartilhada com outros agentes. Os agentes de *intra-slice* utilizam o compartilhamento de parâmetros com o método SAC permitindo que uma única política seja utilizada para todos os agentes de *intra-slice*.

Espaços de Observação: Os espaços de observação representam as entradas dos métodos para os agentes de *inter-slice* e *intra-slice* tendo diferentes composições para cada tipo de agente de escalonamento. O espaço de observação do agente de *inter-slice* no passo de simulação n é definido como

$$\boldsymbol{O}_n^{\text{inter}} = [\boldsymbol{q}_{\text{embb}}, \boldsymbol{q}_{\text{urllc}}, \boldsymbol{q}_{\text{mmtc}}, \boldsymbol{s}_{\text{embb}}, \boldsymbol{s}_{\text{urllc}}, \boldsymbol{s}_{\text{mmtc}}], \tag{28}$$

onde $\boldsymbol{q}_s = [r_{\text{s}}^{\text{req}}, d_{\text{s}}^{\text{req}}, U_s]$ representa os requisitos do *slice* para a taxa de transferência, latência e a quantidade de UEs associados ao *slice*. As métricas de taxa de transferência, ocupação de *buffer* e atraso no *buffer* para cada *slice* são representadas em $\boldsymbol{s}_s = [r_s(n), b_s^{\text{occ}}(n), d_s(n)]$. Essas métricas de *slice* são os valores médios dos valores obtidos dos UEs associados ao *slice* s.

O espaço de observação para o agente de *intra-slice* no passo de simulação n é

$$\boldsymbol{O}_{n,s}^{\text{intra}} = [R_s(n), \boldsymbol{q}_s, \boldsymbol{s}_s], \tag{29}$$

onde, além dos requisitos do *slice* analisado e as suas métricas, também é incluída a quantidade de RBs alocados pelo agente de *inter-slice* para o agente de *intra-slice* em questão.

Espaços de Ação: O espaço de ação do agente de *inter-slice* no passo de simulação n é definido de maneira similar a [63] como $\boldsymbol{A}_n^{\text{inter}} = [a_{\text{embb}}^{\text{inter}}, a_{\text{urllc}}^{\text{inter}}, a_{\text{mmtc}}^{\text{inter}}]$, onde a_s representa um fator de ação para o *slice* s com valor no intervalo $[-1, 1]$. A ação escolhida pelo agente $\boldsymbol{A}_n$ é mapeada para o número de RBs a serem alocados para cada *slice* usando

$$\boldsymbol{A}_n^{\text{RB}} = \left\lfloor \frac{R(\boldsymbol{A}_n + 1)}{\sum_{s \in S}(a_s + 1)} \right\rceil. \tag{30}$$

A ação gerada pelo agente de *inter-slice* define quantos RBs serão utilizados por cada *slice*. O agente de *intra-slice* tem o seu espaço de ação representado por uma única variável a_s^{inter} com valores inteiros no intervalo de $[0, 2]$, onde cada valor representa uma escolha de algoritmo entre as três opções *round-robin*, *maximum-throughput* e *proportional-fair*, que serão responsáveis por alocar os RBs para os UEs.

Funções de Recompensa: De forma similar a [63], a função de recompensa $\mathsf{W}(n)$ considera a distância para cumprir os requisitos de cada *slice* com base na sua construção em conjunto com o uso de mecanismo de proteção ao *slice* prioritário, em que as recompensas dos

slices de eMBB e URLLC são contabilizadas apenas quando os requisitos do *slice* de URLLC são cumpridos. A função de recompensa do agente de *inter-slice* é

$$\mathsf{W}^{\text{inter}}(n) = \begin{cases} \mathsf{W}_{\text{embb}}(n) + \mathsf{W}_{\text{mmtc}}(n), \text{if } \mathsf{W}_{\text{urllc}}(n) = 0 \\ \mathsf{W}_{\text{urllc}}(n), \text{ caso contrário} \end{cases}, \tag{31}$$

onde $\mathsf{W}_{\text{embb}}(n)$, $\mathsf{W}_{\text{urllc}}(n)$ e $\mathsf{W}_{\text{mmtc}}(n)$ representam a função de recompensa para os *slices* de eMBB, URLLC e mMTC no passo de simulação n, respectivamente. Dessa forma, o agente de *inter-slice* tem como prioridade atender aos requisitos do *slice* prioritário de URLLC e, após isso, atender aos requisitos dos *slices* restantes.

O cálculo das contribuições da recompensa de cada *slice* é baseado nos requisitos de taxa de transferência e atraso do *buffer*, sendo dado por

$$\mathsf{W}_{\text{embb}}(n) = -(w^r_{\text{embb}}\mathsf{W}^r_{\text{embb}}(n) + w^d_{\text{embb}}\mathsf{W}^d_{\text{embb}}(n)), \tag{32}$$

$$\mathsf{W}_{\text{urllc}}(n) = -(w^r_{\text{urllc}}\mathsf{W}^r_{\text{urllc}}(n) + w^r_{\text{urllc}}\mathsf{W}^d_{\text{urllc}}(n)), \tag{33}$$

$$\mathsf{W}_{\text{mmtc}}(n) = -w^d_{\text{mmtc}}\mathsf{W}^d_{\text{mmtc}}(n), \tag{34}$$

onde $\mathsf{W}^r_{s\in S}(n)$ e $\mathsf{W}^d_{s\in S}(n))$ representam as contribuições da taxa de transferência e o atraso do *buffer* no cálculo da recompensa. Os pesos w podem definir a prioridade das métricas de cada *slice* e também podem ser utilizados para normalizar os valores da função de recompensa. A contribuição da taxa de transferência é definida como

$$\mathsf{W}^r_s(n) = \begin{cases} \frac{r^{\text{req}}_{\text{s}} - r_{\text{s}}(n)}{r^{\text{req}}_{\text{s}}}, \text{ if } r_{\text{s}}(n) < r^{\text{req}}_{\text{s}} \\ 0, \qquad\quad \text{if } r_{\text{s}}(n) \geq r^{\text{req}}_{\text{s}} \end{cases}, \tag{35}$$

e a contribuição do atraso no *buffer* é

$$\mathsf{W}^d_s(n) = \begin{cases} \frac{d_{\text{s}}(n) - d^{\text{req}}_{\text{s}}}{d_{\max} - d^{\text{req}}_{\text{s}}}, \text{ if } d_{\text{s}}(n) > d^{\text{req}}_{\text{s}} \\ 0, \qquad\quad \text{if } d_{\text{s}}(n) \leq d^{\text{req}}_{\text{s}} \end{cases}. \tag{36}$$

A função de recompensa do agente de *intra-slice* é

$$\mathsf{W}^{\text{intra}}_s(n) = \mathsf{W}^r_s(n) + \mathsf{W}^d_s(n), \tag{37}$$

onde as distâncias para atingir os requisitos das taxa de transferência e do atraso do *buffer* são contabilizadas pelo agente, que busca minimizá-las por meio da seleção, a cada passo de simulação n, do algoritmo de escalonamento mais adequado entre as opções: *round-robin*, *maximum-throughput* e *proportional-fair*. No caso do *slice* de mMTC, o valor da contribuição da taxa de transferência no $\mathsf{W}^r_s(n)$ é sempre considerado zero, dado que o *slice* de mMTC não possui requisitos de taxa de transferência.

9.4 Resultados Numéricos

A Tabela 5 apresenta os principais parâmetros da rede de comunicação empregada na simulação de canal utilizando o simulador QuaDRiGa [68]. O escalonamento de recursos de rádio foi modelado por meio de MARL, utilizando o método SAC, implementado com o suporte da biblioteca Ray Rllib [69]. O esquema proposto é composto por um agente de escalonamento *inter-slice* e três agentes *intra-slice*, os quais selecionam entre os algoritmos *round-robin*,

Tabela 5: Parâmetros de Simulação do Escalonador MARL Proposto.

Parâmetro	**Valor**
Frequência da portadora	6 GHz
Número de RBs	100
Subcarrier spacing	15 kHz
Duração do TTI	1 ms
Potência de transmissão	35 dBm
Velocidade dos UEs	3 km/h
Número de UEs	100
Número de *slices*	3
Rodadas de simulação	100
TTI	1000
$r^{\text{req}}_{\text{embb}}$, $r^{\text{req}}_{\text{urllc}}$	20 Mbps, 5 Mbps
$d^{\text{req}}_{\text{embb}}$, $d^{\text{req}}_{\text{urllc}}$, $d^{\text{req}}_{\text{mmtc}}$	30 ms, 1 ms, 50 ms
μ_{embb}, μ_{urllc}, μ_{mmtc}	20 Mbps, 5 Mbps, 0.1 Mbps
w^{r}_{embb}, w^{d}_{embb}, w^{r}_{urllc}, w^{d}_{urllc}, w^{r}_{mmtc}	0.5, 0.3, 0.5, 0.5, 0.2

maximum-throughput e *proportional-fair*. Para fins de comparação, foi utilizado como método *baseline* o escalonador *Satisfaction Rate* (SSR), conforme descrito em [63].

O cenário considerado neste trabalho é um galpão de fábrica, com uma BS equipada com múltiplas antenas e posicionada no canto superior. A BS atende UEs em três casos de uso distintos: eMBB, URLLC e mMTC. Para aumentar o realismo do cenário, a simulação foi baseada em medições reais feitas em um galpão de fábrica localizado em Nuremberg, Alemanha, que extraímos de [65]. Neste cenário, a BS fornece serviço para um total de 100 UEs, distribuídos uniformemente por todo o galpão da fábrica. Dentre esses UEs, 30 estão associados com eMBB, 40 com URLLC e 30 com aplicações mMTC. Foi considerado 100 RBs para distribuição entre os *slices* e UEs, utilizando tanto escalonamento *inter-slice* quanto *intra-slice*, com potência alocada uniformemente entre os RBs. Além disso, assume-se que os UEs estão em movimento a uma velocidade de 3 km/h. O sistema apresenta três *slices*, sendo uma dedicada ao eMBB, uma ao URLLC e uma ao mMTC. Por fim, as 100 rodadas de simulação geradas pelo QuaDRiGa foram divididas em dois subconjuntos distintos: um conjunto destinado ao treinamento e outro reservado para testes. O conjunto de treinamento compreende 70 rodadas, enquanto o conjunto de teste consiste em 30 rodadas.

A Figura 25 mostra os resultados para as taxas de transferência médias para cada *slice*. Considerando os requisitos para taxas de transferência $r^{\text{req}}_{\text{embb}} = 20$ e $r^{\text{req}}_{\text{urllc}} = 5$ Mbps, o agente proposto forneceu uma taxa de transferência maior para o *slice* de URLLC e uma taxa menor para o eMBB quando comparado com o método SSR. Esse comportamento é justificado pois o método proposto primeiramente atende os requisitos do *slice* prioritário e, posteriormente, aloca os recursos restantes para o *slice* de eMBB. Tanto o método proposto quanto o método SSR não conseguiram cumprir o requisito de taxa de transferência para o *slice* de eMBB, pois a quantidade de recursos disponíveis na rede não era suficiente para atender a todos os requisitos.

Quando se avalia o atraso no *buffer*, a Figura 26 mostra que o método proposto conseguiu manter o atraso do *slice* de URLLC abaixo do requisito de $d^{\text{req}}_{\text{urllc}} = 1$ ms, enquanto o método SSR manteve atraso acima de 1 ms por pelo menos 50% dos passos de simulação. Ambos os métodos cumpriram o requisito do *slice* eMBB $d^{\text{req}}_{\text{embb}} = 30$ ms. O método proposto conseguiu

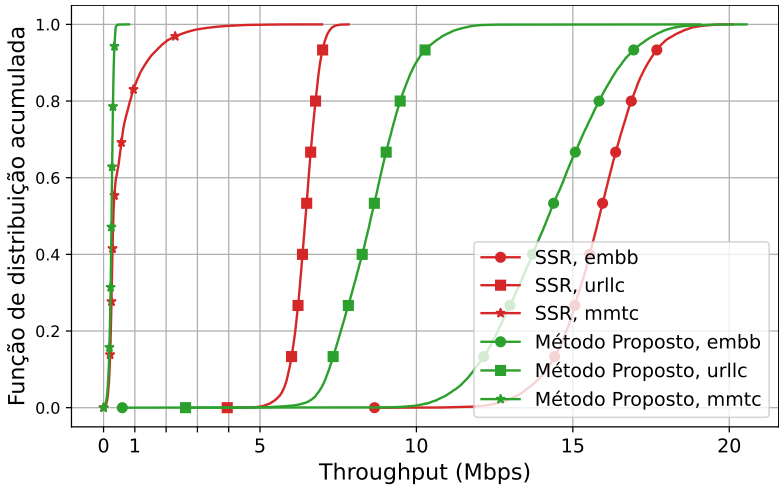


Figura 25: Taxas de transferência média obtidas por cada *slice*.

cumprir os requisitos de latência para o mMTC de $d^{\text{req}}_{\text{mmtc}} = 50$ ms durante 80% do período da simulação, enquanto o método de SSR conseguiu cumprir durante toda a simulação.

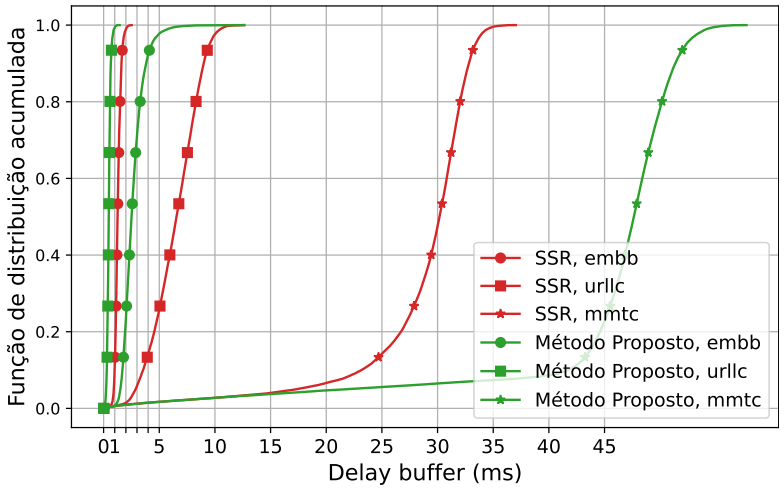


Figura 26: Atraso médio do *buffer* experienciados por cada *slice*.

Por fim, a Figura 27 mostra os valores médios de violações de requisitos totais e para o *slice* prioritário URLLC em 30 episódios de teste. O método proposto apresenta um menor número de violações considerando-se todos os *slices* e também o *slice* prioritário. O número de violações para o slice prioritário URLLC mantém valores próximos a zero durante toda a simulação e com um pequeno desvio padrão se comparado aos valores obtidos para o URLLC pelo SSR.

9.5 Conclusão

Foi apresentado um método de alocação de recursos de rádio baseado em MARL, o qual executa as funções de escalonamento *inter-slice* e *intra-slice*, com foco em cenários típicos da Indústria 4.0. O agente proposto cumpre os requisitos dos *slices* através da combinação de ações do escalonador de *inter-slice* e *intra-slice*, além de priorizar os *slices* de URLLC quando os recursos disponíveis na rede não são suficientes para atender os requisitos de todos os *slices*.

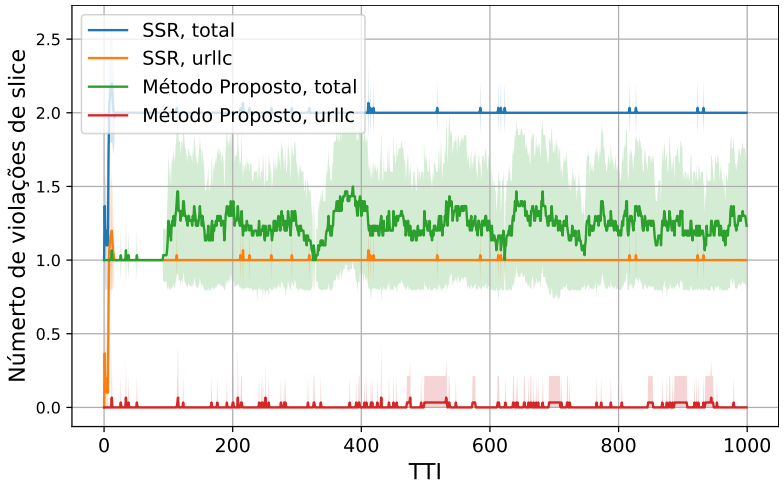


Figura 27: Média de violações e desvio padrão nos dados de teste.

Os resultados preliminares mostram que o método proposto obteve um número de violações significativamente menor em relação ao método de SSR nos *slices* gerais e prioritários.

10 Conclusão

Este relatório consolidou os principais avanços obtidos na Atividade 3.3 da Fase III do projeto Brasil 6G, com foco no desenvolvimento de soluções inovadoras voltadas a aplicações de IoT, em alinhamento com os requisitos emergentes das redes 6G. Os trabalhos realizados durante o primeiro ano desta fase demonstram contribuições relevantes em áreas estratégicas como *offloading* de tarefas em borda para realidade aumentada, codificação semântica de vídeo para redes com restrições, alocação de recursos em jogos em nuvem com realidade virtual, carregamento sem fio eficiente e seguro, posicionamento *indoor* com alta acurácia, comunicações robustas em ambientes com desvanecimento complexo, vigilância ambiental inteligente e orquestração dinâmica de recursos em ambientes industriais.

As soluções propostas exploram tecnologias avançadas como aprendizado profundo, comunicação semântica, controle adaptativo de qualidade de serviço, sistemas ciberfísicos, redes inteligentes e aprendizado por reforço multiagente, evidenciando a abordagem multidisciplinar necessária para o desenvolvimento do ecossistema 6G. Os resultados obtidos contribuem para o amadurecimento de técnicas e arquiteturas capazes de atender aos desafios impostos por aplicações críticas, imersivas e de larga escala, promovendo conectividade mais eficiente, inteligente e confiável.

Dessa forma, os estudos aqui apresentados não apenas reforçam a viabilidade das tecnologias investigadas, mas também abrem caminho para sua futura integração em sistemas reais, com potencial impacto em diversos setores, como indústria, saúde, agricultura, cidades inteligentes e preservação ambiental. As próximas etapas do projeto deverão aprofundar a validação em cenários reais e a exploração de mecanismos de interoperabilidade, escalabilidade e segurança, fundamentais para a consolidação de soluções 6G orientadas à IoT.

Referências

[1] J. Cao, K.-Y. Lam, L.-H. Lee, X. Liu, P. Hui, and X. Su, "Mobile Augmented Reality: User Interfaces, Frameworks, and Intelligence," *ACM Comput. Surv.*, vol. 55, no. 9, 2023.

[2] T. Masood and J. Egger, "Augmented reality in support of Industry 4.0: Implementation challenges and success factors," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 181–195, 2019.

[3] C. Moro, C. Phelps, P. Redmond, and Z. Stromberga, "HoloLens and mobile augmented reality in medical and health science education: A randomised controlled trial," *British Journal of Educational Technology*, vol. 52, no. 2, pp. 680–694, 2021.

[4] V. Yadhav, A. Williams, O. Smid, J. Kjällman, R. Islam, J. Halén, and W. John, "Dynamic Computational Offloading for Mobile Devices," in *Proceedings of the 14th International Conference on Cloud Computing and Services Science - CLOSER*, ser. CLOSER'24. SciTePress, 2024, pp. 265—-276.

[5] K. Toczé *et al.*, "Performance Study of Mixed Reality for Edge Computing," in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, 2019, p. 285–294.

[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[7] F. Muzzini, N. Capodieci, R. Cavicchioli, and B. Rouxel, "Brief Announcement: Optimized GPU-accelerated Feature Extraction for ORB-SLAM Systems," in *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures*, ser. SPAA '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3558481.3591310

[8] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, "MIS-SLAM: Real-Time Large-Scale Dense Deformable SLAM System in Minimal Invasive Surgery Based on Heterogeneous Computing," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4068–4075, 2018.

[9] D. G. Morín, P. Pérez, and A. G. Armada, "Toward the Distributed Implementation of Immersive Augmented Reality Architectures on 5G Networks," *IEEE Communications Magazine*, vol. 60, no. 2, pp. 46–52, 2022.

[10] N. Blum, S. Lachapelle, and H. Alvestrand, "WebRTC: Real-time communication for the open web platform," *Communications of the ACM*, vol. 64, no. 8, pp. 50–54, 2021.

[11] D. Rico and P. Merino, "A survey of end-to-end solutions for reliable low-latency communications in 5G networks," *IEEE Access*, vol. 8, pp. 192 808–192 834, 2020.

[12] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 210–219, 2022.

[13] L. Xia, Y. Sun, C. Liang, D. Feng, R. Cheng, Y. Yang, and M. A. Imran, "WiserVR: Semantic communication enabled wireless virtual reality delivery," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 32–39, 2023.

[14] C. Liang, X. Deng, Y. Sun, R. Cheng, L. Xia, D. Niyato, and M. A. Imran, "VISTA: Video Transmission over A Semantic Communication Approach," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2023, pp. 1777–1782.

[15] S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, and P. Zhang, "Wireless deep video semantic transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 214–229, 2023.

[16] V. Sivaraman, P. Karimi, V. Venkatapathy, M. Khani, S. Fouladi, M. Alizadeh, F. Durand, and V. Sze, "Gemino: Practical and robust neural compression for video conferencing," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 569–590.

[17] T. Li, V. Sivaraman, P. Karimi, L. Fan, M. Alizadeh, and D. Katabi, "Reparo: Loss-Resilient Generative Codec for Video Conferencing," *arXiv preprint arXiv:2305.14135*, 2023.

[18] Y. Cheng, Z. Zhang, H. Li, A. Arapin, Y. Zhang, Q. Zhang, Y. Liu, K. Du, X. Zhang, F. Y. Yan *et al.*, "{GRACE}:{Loss-Resilient}{Real-Time} video through neural codecs," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, 2024, pp. 509–531.

[19] Y. Wen, Z. Zhang, J. Sun, J. Li, C. S. Chen, and G. Niu, "SAW: Semantic-Aware WebRTC Transmission Using Diffusion-Based Scalable Video Coding," *IEEE Internet of Things Journal*, 2024.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[21] M. Shokrnezhad, H. Yu, T. Taleb, R. Li, K. Lee, J. Song, and C. Westphal, "Towards a Dynamic Future with Adaptable Computing and Network Convergence (ACNC)," 2024. [Online]. Available: https://arxiv.org/abs/2403.07573

[22] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Study on Network Controlled Interactive Service (NCIS) in the 5G System (Release 17)," 3GPP, Tech. Rep. TR 22.842 V17.2.0, 2019.

[23] ——, "3rd Generation Partnership Project; Technical Specification Group RAN; Study on XR evaluations for NR (Release 17)," 3GPP, Tech. Rep. TR 38.838 V17.0.0, 2022.

[24] ——, "3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; XR in 5G (Release 18)," 3GPP, Tech. Rep. TR 22.842 V18.0.0, 2023.

[25] Huawei, "Huawei; iLab; Cloud VR Network Solution White Paper (2018)," Huawei, Tech. Rep., 2018.

[26] C. Baena, S. Fortes, O. S. Penaherrera-Pulla, E. Baena, and R. Barco, "Gaming in the Cloud: 5G as the Pillar for Future Gaming Approaches," *IEEE Communications Magazine*, pp. 1–7, 2024.

[27] H. Du, J. Liu, D. Niyato, J. Kang, Z. Xiong, J. Zhang, and D. I. Kim, "Attention-Aware Resource Allocation and QoE Analysis for Metaverse xURLLC Services," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 7, pp. 2158–2175, 2023.

[28] J. P. Esper *et al.*, "Impact of User Privacy and Mobility on Edge Offloading," in *IEEE International Symposium on PIMRC*, 2023, pp. 1–6.

[29] O. L. A. López, H. Alves, R. D. Souza, S. Montejo-Sánchez, E. M. G. Fernández, and M. Latva-Aho, "Massive Wireless Energy Transfer: Enabling Sustainable IoT Toward 6G Era," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8816–8835, 2021.

[30] L. Ginting, H. S. Yoon, D. I. Kim, and K. W. Choi, "Beam Avoidance for Human Safety in Radiative Wireless Power Transfer," *IEEE Access*, vol. 8, pp. 217 510–217 525, 2020.

[31] B. Clerckx, K. Huang, L. R. Varshney, S. Ulukus, and M.-S. Alouini, "Wireless Power Transfer for Future Networks: Signal Processing, Machine Learning, Computing, and Sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 5, pp. 1060–1094, 2021.

[32] O. L. A. López, D. Kumar, R. D. Souza, P. Popovski, A. Tölli, and M. Latva-Aho, "Massive MIMO With Radio Stripes for Indoor Wireless Energy Transfer," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 7088–7104, 2022.

[33] A. Azarbahram, O. L. A. López, and M. Latva-Aho, "Waveform Optimization and Beam Focusing for Near-Field Wireless Power Transfer With Dynamic Metasurface Antennas and Non-Linear Energy Harvesters," *IEEE Transactions on Wireless Communications*, vol. 24, no. 2, pp. 1031–1045, 2025.

[34] J. Zhou, P. Zhang, J. Han, L. Li, and Y. Huang, "Metamaterials and Metasurfaces for Wireless Power Transfer and Energy Harvesting," *Proceedings of the IEEE*, vol. 110, no. 1, pp. 31–55, 2022.

[35] X. Guo, N. Ansari, F. Hu, Y. Shao, N. R. Elikplim, and L. Li, "A Survey on Fusion-Based Indoor Positioning," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 566–594, Nov. 2020, doi: 10.1109/COMST.2019.2951036.

[36] J. A. López-Pastor, A. J. Ruiz-Ruiz, A. S. Martínez-Sala, and J. Luis Gómez-Tornero, "Evaluation of an indoor positioning system for added-value services in a mall," in *2019 Int. Conf. on Indoor Positioning and Indoor Navig. (IPIN)*, Nov. 2019, pp. 1–8, doi: 10.1109/IPIN.2019.8911822.

[37] K. Casareo and Z. Chaczko, "Beacon-Based Localization Middleware for Tracking in Medical and Healthcare Environments," in *2018 12th Int. Symp. on Med. Inf. and Commun. Technol. (ISMICT)*, Dec. 2018, pp. 1–6, doi: 10.1109/ISMICT.2018.8573701.

[38] P. S. Farahsari, A. Farahzadi, J. Rezazadeh, and A. Bagheri, "A Survey on Indoor Positioning Systems for IoT-Based Applications," vol. 9, pp. 7680–7699, Feb. 2022, doi: 10.1109/JIOT.2022.3149048.

[39] K. Lam, "Bluetooth 5.1 Direction Finding: Theory and Practice," Bluetooth Special Interest Group (SIG), Tech. Rep., May 2019. [Online]. Available: https://tinyurl.com/lam2019DF

[40] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. of the ASME – J. of Basic Eng.*, vol. 82, no. Series D, pp. 35–45, 1960, doi: 10.1115/1.3662552.

[41] M. Girolami, F. Furfari, P. Barsocchi, and F. Mavilia, "A Bluetooth 5.1 Dataset Based on Angle of Arrival and RSS for Indoor Localization," vol. 11, pp. 81 763–81 776, Aug. 2023, doi: 10.1109/ACCESS.2023.3301126.

[42] Y. Bar-Shalom and L. Campo, "The Effect of the Common Process Noise on the Two-Sensor Fused-Track Covariance," vol. AES-22, no. 6, pp. 803–805, Nov. 1986, doi: 10.1109/TAES.1986.310815.

[43] J. Gao and C. Harris, "Some remarks on Kalman filters for the multisensor fusion," *Inf. Fusion*, vol. 3, no. 3, pp. 191–201, Sept. 2002, doi: 10.1016/S1566-2535(02)00070-2.

[44] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep Reinforcement Learning-Based Channel Allocation for Wireless LANs with Graph Convolutional Networks," 2019. [Online]. Available: https://arxiv.org/abs/1905.07144

[45] S. R. Doha and A. Abdelhadi, "Deep Learning in Wireless Communication Receiver: A Survey," 2025. [Online]. Available: https://arxiv.org/abs/2501.17184

[46] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[47] H.-Y. Chen, M.-H. Wu, T.-W. Yang, C.-W. Huang, and C.-F. Chou, "Attention-Aided Autoencoder-Based Channel Prediction for Intelligent Reflecting Surface-Assisted Millimeter-Wave Communications," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 4, pp. 1906–1919, 2023.

[48] Z. Huang, D. He, J. Chen, Z. Wang, and S. Chen, "Autoencoder with fitting network for Terahertz wireless communications: A deep learning approach," *China Communications*, vol. 19, no. 3, pp. 172–180, 2022.

[49] H.-P. Yin, X.-H. Zhao, J.-L. Yao, and H.-P. Ren, "Deep-Learning-Based Channel Estimation for Chaotic Wireless Communication," *IEEE Wireless Communications Letters*, vol. 13, no. 1, pp. 143–147, 2024.

[50] H. Safi, I. Tavakkolnia, and H. Haas, "Deep Learning Based End-to-End Optical Wireless Communication Systems With Autoencoders," *IEEE Communications Letters*, vol. 28, no. 6, pp. 1342–1346, 2024.

[51] M. D. Yacoub, "The κ-μ distribution and the η-μ distribution," vol. 49, no. 1, pp. 68–81, 2007.

[52] J. Verde and J. L. Zêzere, "Assessment and validation of wildfire susceptibility and hazard in Portugal," *Natural Hazards and Earth System Sciences*, vol. 10, no. 3, pp. 485–497, 2010.

[53] M. Flannigan, A. S. Cantin, W. J. De Groot, M. Wotton, A. Newbery, and L. M. Gowman, "Global wildland fire season severity in the 21st century," *Forest Ecology and Management*, vol. 294, pp. 54–61, 2013.

[54] J. F. Santos, R. V. Soares, and A. C. Batista, "Perfil dos incêndios florestais no brasil em áreas protegidas no período de 1998 a 2002," *Floresta*, vol. 36, no. 1, pp. 93–100, 2006.

[55] W. J. Franca Rocha, R. N. Vasconcelos, S. G. Duverger, D. P. Costa, N. A. Santos, R. O. Franca Rocha, M. M. de Santana, A. A. Alencar, V. L. Arruda, W. V. d. Silva *et al.*, "Mapping Burned Area in the Caatinga Biome: Employing Deep Learning Techniques," *Fire*, vol. 7, no. 12, p. 437, 2024.

[56] M. M. Veras, "Mudanças climáticas e incêndios florestais: implicações sobre a saúde," *Ciência e Cultura*, vol. 76, no. 3, pp. 01–07, 2024.

[57] Y. Liu *et al.*, "Network Slicing for eMBB, URLLC, and mMTC: An Uplink Rate-Splitting Multiple Access Approach," *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 2140–2152, Mar. 2024.

[58] F. Debbabi *et al.*, "An Overview of Interslice and Intraslice Resource Allocation in B5G Telecommunication Networks," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 5120–5132, Dec. 2022.

[59] M. Zangooei *et al.*, "Reinforcement Learning for Radio Resource Management in RAN Slicing: A Survey," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 118–124, Feb. 2023.

[60] Y. Shao *et al.*, "Graph Attention Network-Based Multi-Agent Reinforcement Learning for Slicing Resource Management in Dense Cellular Network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 792–10 803, Oct. 2021.

[61] M. Zangooei *et al.*, "Flexible RAN Slicing in Open RAN With Constrained Multi-Agent Reinforcement Learning," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2, pp. 280–294, Feb. 2024.

[62] H. Anil Akyildiz *et al.*, "Hierarchical Reinforcement Learning Based Resource Allocation for RAN Slicing," *IEEE Access*, vol. 12, pp. 75 818–75 831, May 2024.

[63] C. Nahum *et al.*, "Safeguard URLLC Services in Industry 4.0 Through Reinforcement Learning Scheduling," in *2023 Workshop on Communication Networks and Power Systems (WCNPS)*. IEEE, 2023, pp. 1–7.

[64] 3GPP, "TS 38.873 v12.5.0," *Study on 3D channel model for LTE*, 2017.

[65] S. Jaeckel *et al.*, "Industrial Indoor Measurements from 2-6 GHz for the 3GPP-NR and QuaDRiGa Channel Model," in *IEEE 90th Vehicular Technology Conference*, 2019, pp. 1–7.

[66] C. V. Nahum *et al.*, "Intent-Aware Radio Resource Scheduling in a RAN Slicing Scenario Using Reinforcement Learning," *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 2253–2267, Mar. 2024.

[67] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. of International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[68] S. Jaeckel *et al.*, "QuaDRiGa - quasi deterministic radio channel generator, user manual and documentations," 2021.

[69] E. Liang *et al.*, "Ray rllib: A composable and scalable reinforcement learning library," *arXiv preprint arXiv:1712.09381*, vol. 85, p. 245, 2017.