UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO (PPGCC)

CLEYBER BEZERRA DOS REIS

SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

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

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

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

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

1. Identificação do material bibliográfico

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

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

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

2. Nome completo do autor

Cleyber Bezerra dos Reis

3. Título do trabalho

SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G

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

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

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

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

Documento assinado eletronicamente por **Antonio Carlos De Oliveira Junior**, **Professor do Magistério Superior**, em 28/10/2025, às 18:00, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Cleyber Bezerra Dos Reis**, **Discente**, em 31/10/2025, às 10:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5748405** e o código CRC **A69DBFD4**.

Referência: Processo nº 23070.049066/2025-36 SEI nº 5748405

CLEYBER BEZERRA DOS REIS

SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G

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

Área de concentração: Ciência da Computação
Linha de Pesquisa: Sistemas de Computação

Orientador: Prof. Dr. Antonio Carlos de Oliveira Júnior
Coorientadora: Profa. Dra. Maria do Rosário Campos Ribeiro

Goiânia
2025

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

REIS, CLEYBER BEZERRA DOS
SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G [manuscrito] / CLEYBER BEZERRA DOS REIS. - 2025.
90 f.

Orientador: Prof. Dr. ANTONIO CARLOS DE OLIVEIRA JÚNIOR; co-orientadora Dra. MARIA DO ROSARIO CAMPOS RIBEIRO.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2025.
Bibliografia. Apêndice.
Inclui siglas, abreviaturas, gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Aprendizado Dividido. 2. ns-3 (5G-LENA, ns3-ai). 3. Redes Móveis 5G/B5G. 4. Métricas de Rede. 5. Rede Neural Convulacional (CNN). I. JÚNIOR, ANTONIO CARLOS DE OLIVEIRA, orient. II. Título.

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **20** da sessão de Defesa de Dissertação de **Cleyber Bezerra dos Reis**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos dois dias do mês de outubro de dois mil e vinte e cinco, a partir das nove horas e trinta minutos, na sala 250 do INF, realizou-se a sessão pública de Defesa de Dissertação intitulada **"SLArch: A Network Metric-aware Split Learning Architecture for B5G/6G Mobile Networks"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor Antonio Carlos de Oliveira Júnior (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professora Doutora Maria do Rosário Campos Ribeiro (INF/UFG & INESC-TEC CRACS), coorientadora; Professor Doutor Waldir Aranha Moreira Júnior (Fraunhofer Portugal AICOS), membro titular externo; Professor Doutor Victor Hugo Lázaro Lopes (IFG), membro titular externo. A participação do Professor Doutor Waldir Aranha Moreira Júnior e do Professor Doutor Victor Hugo Lázaro Lopes ocorreu por meio de videoconferência. Durante a arguição os membros da banca sugeriram a alteração do título para a lingua portuguesa. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Antonio Carlos de Oliveira Júnior, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos dois dias do mês de outubro de dois mil e vinte e cinco.

TÍTULO SUGERIDO PELA BANCA

SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G

Documento assinado eletronicamente por **Victor Hugo Lázaro Lopes**, **Usuário Externo**, em 02/10/2025, às 11:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Maria Do Rosario Campos Ribeiro**, **Professor do Magistério Superior-Substituto**, em 02/10/2025, às 11:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Waldir Aranha Moreira Junior**, **Usuário Externo**, em 02/10/2025, às 11:47, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Antonio Carlos De Oliveira Junior**, **Professor do Magistério Superior**, em 02/10/2025, às 11:48, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

Documento assinado eletronicamente por **Cleyber Bezerra Dos Reis**, **Discente**, em 02/10/2025, às 13:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

CLEYBER BEZERRA DOS REIS

SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 02 de Outubro de 2025, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Antonio Carlos de Oliveira Júnior
Instituto de Informática – UFG
Presidente da Banca

Profa. Dra. Maria do Rosário Campos Ribeiro
Instituto de Informática – UFG

Prof. Dr. Victor Hugo Lázaro Lopes
Instituto Federal de Goiás – IFG

Prof. Dr. Waldir Aranha Moreira Júnior
Fraunhofer Portugal – AICOS

Cleyber Bezerra dos Reis

O autor é graduado em Análise e Desenvolvimento de Sistemas pela Universidade Salgado de Oliveira (2003) e Especialista em Redes de Computadores e Segurança de Sistemas pela UFG (2021). Concursado como Técnico em Eletrônica Pleno na Empresa Brasileira de Correios e Telégrafos (GO), atuando em redes, fiscalização de contratos e suporte técnico. Interesses: Teleinformática, Redes de Computadores, Automação, IA e IoT.

Dedico esta dissertação ao Altíssimo Deus, fonte de vida e de toda sabedoria, por iluminar meu caminho, fortalecer minha fé e conceder-me discernimento nas horas de incerteza.

À minha esposa, Maria Luiza Gonçalves André Silva Bezerra, cuja força, amor e zelo cotidiano sustentaram-me nas horas de maior desafio e tornaram leve o caminho até esta conquista; às minhas filhas, Debborah Gonçalves Bezerra e Alyne Gonçalves Bezerra, que enchem meus dias de sentido, esperança e orgulho.

Aos meus pais, João Reis de Sousa e Teresa Bizerra de Sousa, exemplos de integridade e dedicação, que me ensinaram o valor do esforço e da perseverança.

A todos que, de alguma forma, acreditaram nesta caminhada e partilharam comigo o sonho que aqui se concretiza.

Agradecimentos

A Deus, pela fortaleza, pela orientação e pelas condições concedidas para que fosse possível concluir mais esta etapa da minha trajetória acadêmica e profissional. À minha esposa, às minhas filhas e aos meus pais, pelo amor incondicional, pelo apoio permanente e pela compreensão diante das inúmeras exigências deste percurso. Ao meu orientador, professor Antonio Carlos de Oliveira Junior, e à minha coorientadora, professora Maria do Rosário Campos Ribeiro, pela confiança depositada, pelos ensinamentos, pela paciência e pelas contribuições essenciais ao desenvolvimento desta pesquisa. Ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Goiás, pela oportunidade de formação científica e pelo ambiente acadêmico que possibilitou a realização deste trabalho. Ao Laboratório de Redes e Arquitetura de Alto Desempenho (LABORA/INF), pelo espaço de pesquisa, pelo suporte técnico e pelas discussões que enriqueceram este estudo. À banca avaliadora, pelo rigor acadêmico, pela disponibilidade e pelas valiosas contribuições que ampliaram a qualidade desta dissertação. Aos professores do curso, pela dedicação no ensino e pela formação sólida transmitida ao longo da minha jornada. À Empresa Brasileira de Correios e Telégrafos, pela oportunidade concedida por meio do teletrabalho, que viabilizou a conciliação entre atividades profissionais e a pesquisa acadêmica. Aos amigos e colegas, pela parceria construtiva, pelo estímulo constante e pela colaboração indispensável ao longo deste processo.

"Sabedoria é a coisa principal: adquire sabedoria e todo o resto virá."

Provérbios 4:7

Resumo

Reis, Cleyber Bezerra. **SLArch: Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G**. Goiânia, 2025. 95p. Dissertação de Mestrado. Programa de Pós-Graduação em Ciência da Computação (PPGCC), Instituto de Informática, Universidade Federal de Goiás.

O *Split Learning* (SL) é um paradigma de aprendizado colaborativo no qual um modelo de rede neural é particionado entre cliente e servidor, possibilitando treinamento distribuído sem a necessidade de compartilhar os dados originais. Nesta dissertação, investigamos uma arquitetura de SL sensível às condições de rede, desenvolvida no simulador *Network Simulator 3* (`ns-3`) com o módulo `5G-LENA` e a interface `ns3-ai`, a fim de avaliar a viabilidade desse paradigma em cenários *Beyond 5G (B5G)* e *6G*. O *framework* proposto, denominado *Split-Learning-ns3*, constitui uma infraestrutura aberta, reprodutível e extensível, capaz de integrar processos de aprendizado de máquina a dinâmicas realistas de rede, considerando métricas como latência, vazão, *jitter*, taxa de perda de pacotes (PLR) e consumo energético. Para validar a proposta, estabelecemos um protocolo experimental com redes neurais convolucionais (CNN) treinadas sobre o conjunto de dados *MNIST*, distribuídas entre diferentes fatiamentos (URLLC, eMBB, mMTC) sob múltiplas numerologias e configurações de parte de largura de banda (BWP). Os resultados mostram que a PLR é o fator dominante para a convergência, enquanto latência e vazão exercem influência moderada; contudo, o *jitter* e o consumo energético apresentam papéis mensuráveis, porém secundários. A análise por fatiamento confirma que o URLLC assegura menor latência, o eMBB concentra maior vazão e o mMTC demonstra maior eficiência energética. Esses achados reforçam a importância da confiabilidade do enlace e da alocação de recursos para o desempenho do SL em condições realistas. Além de fornecer uma base metodológica reprodutível, esta dissertação oferece evidências empíricas para o treinamento distribuído resiliente e energeticamente eficiente na borda da rede sem fio (*wireless edge*), com aplicações potenciais em setores críticos como saúde, transporte autônomo e Indústria 4.0.

Palavras–chave

Aprendizado Dividido, ns-3 (5G-LENA, ns3-ai), Redes Móveis B5G/6G, Métricas de Rede, Rede Neural Convulacional (CNN).

Abstract

Reis, Cleyber Bezerra. **SLArch: A Network Metric-aware Split Learning Architecture for B5G/6G Mobile Networks**. Goiânia, 2025. 95p. MSc. Dissertation. Programa de Pós-Graduação em Ciência da Computação (PPGCC), Instituto de Informática, Universidade Federal de Goiás.

Split Learning (SL) is a collaborative learning paradigm in which a neural network model is partitioned between client and server, enabling distributed training without the need to share raw data. In this dissertation, we investigate a network-aware SL architecture developed in the Network Simulator 3 (ns-3) with the 5G-LENA module and the ns3-ai interface, in order to assess the feasibility of this paradigm in Beyond 5G (B5G) and 6G scenarios. The proposed framework, named Split-Learning-ns3, provides an open, reproducible, and extensible infrastructure capable of integrating machine learning processes into realistic network dynamics, considering metrics such as latency, throughput, jitter, packet loss ratio (PLR), and energy consumption. To validate the proposal, we designed an experimental protocol with convolutional neural networks (CNNs) trained on the MNIST dataset, distributed across different network slices (Ultra-Reliable Low-Latency Communications – URLLC, Enhanced Mobile Broadband – eMBB, and massive Machine Type Communications – mMTC) under multiple numerologies and bandwidth part (BWP) configurations. The results indicate that PLR is the dominant factor for convergence, whilst latency and throughput exert a moderate influence; jitter and energy consumption play measurable yet secondary roles. The slice-based analysis confirms that URLLC ensures lower latency, eMBB delivers higher throughput, and mMTC achieves greater energy efficiency. These findings highlight the importance of link reliability and resource allocation for SL performance under realistic conditions. Beyond providing a reproducible methodological basis, this dissertation offers empirical evidence towards resilient and energy-efficient distributed training at the wireless edge, with potential applications in critical sectors such as healthcare, autonomous transport, and Industry 4.0.

Keywords

Split Learning, ns-3 (5G-LENA, ns3-ai), B5G/6G Mobile Networks, Network Metrics, Convolutional Neural Network (CNN).

Sumário

Lista de Algoritmos

Lista de Figuras

Lista de Tabelas

Lista de Siglas

3GPP *Third Generation Partnership Project*.
4G *Fourth Generation Mobile Networks (IMT-Advanced)*.
5G *Fifth Generation Mobile Networks*.
5G NR *5G New Radio*.
5G-LENA *5G Link-Level Emulator for Network Analysis*.
5QI *5G QoS Identifier*.
6G *Sixth Generation Mobile Networks*.

AI *Artificial Intelligence*.
AMF *Access and Mobility Management Function*.
ANOVA *Analysis of Variance*.
AR *Augmented Reality*.

B5G *Beyond Fifth Generation 5G*.
B5G/6G *Beyond Fifth or Sixth Generation Wireless Networks*.
BWP *Bandwidth Part*.

CG *Configured Grant*.
CIFAR-10 *Canadian Institute For Advanced Research — 10 classes*.
CNN *Convolutional Neural Network*.
CP *Cyclic Prefix*.
CP *Control Plane*.
CQI *Channel Quality Indicator*.
CTTC *Centre Tecnològic de Telecomunicacions de Catalunya*.

DL *Deep Learning*.
DRB *Data Radio Bearer*.
DRL *Deep Reinforcement Learning*.

EDGE *Enhanced Data Rates for GSM Evolution*.
eMBB *enhanced Mobile Broadband*.
EPC *Evolved Packet Core*.
EPSL *Efficient Parallel Split Learning*.

FDD *Frequency Division Duplex*.
FDR *False Discovery Rate*.
FL *Federated Learning*.
FWA *Fixed Wireless Access*.

gNB *Next Generation NodeB*.

IC *Confidence Interval*.
IEEE *Institute of Electrical and Electronics Engineers*.
IoT *Internet of Things*.
IP *Internet Protocol*.

MAC *Media Access Control*.
MIMO *Multiple-Input and Multiple-Output*.
ML *Machine Learning*.
mMTC *massive Machine-Type Communications*.
MNIST *Modified National Institute of Standards and Technology*.

NAS *Non-Access Stratum*.
NS *Network Slicing*.
ns-3 *Network Simulator 3*.
ns-O-RAN *ns-3 Open RAN integration*.
ns3-ai *ns-3 and AI Integration Framework*.

O-RAN *Open Radio Access Network*.
OFDM *Orthogonal Frequency Division Multiplexing*.

PCF *Policy Control Function*.
PLR *Packet Loss Rate*.
PSL *Parallel Split Learning*.

QoS *Quality of Service*.

RAN *Radio Access Network*.
RIC *RAN Intelligent Controller*.
RL *Reinforcement Learning*.
RRC *Radio Resource Control*.

SFL *Split Federated Learning*.
SGD *Stochastic Gradient Descent*.
SL *Split Learning*.
SMF *Session Management Function*.

SPS *Semi-Persistent Scheduling*.

TDD *Time Division Duplex*.

UAV *Unmanned Aerial Vehicle*.
UE *User Equipment*.
UP *User Plane*.
URLLC *Ultra-Reliable Low-Latency Communications*.

V2X *Vehicle-to-Everything*.
VR *Virtual Reality*.

XR *Extended Reality*.

CAPÍTULO 1

Introdução

A *Artificial Intelligence* (AI) consolidou-se como um elemento estruturante em diferentes setores da sociedade contemporânea. Hoje, ela sustenta aplicações críticas em saúde, sistemas de transporte autônomos e processos industriais, nas quais decisões rápidas e confiáveis são indispensáveis. O avanço do aprendizado profundo e das técnicas colaborativas de treinamento ampliou horizontes, mas também evidenciou limitações relativas à privacidade de dados, ao consumo energético e à dependência de infraestrutura de comunicação. Nesse cenário, paradigmas como o *Split Learning* (SL) despontam como alternativas promissoras, sobretudo quando aplicados a redes móveis de próxima geração. Ao mesmo tempo, o *Network Slicing* (NS) emerge como pilar no *Fifth Generation Mobile Networks* (5G) e *Beyond Fifth Generation 5G* (B5G), por permitir a coexistência de múltiplos perfis de tráfego (*Ultra-Reliable Low-Latency Communications* (URLLC), *enhanced Mobile Broadband* (eMBB), *massive Machine-Type Communications* (mMTC)) com requisitos heterogêneos de latência, confiabilidade e vazão [Popovski et al. 2018, 3GPP TS 23.501 2025]. A interação entre SL e NS torna-se, assim, um desafio estratégico: compreender como cada perfil molda a acurácia, a convergência e a robustez do treinamento distribuído, ao mesmo tempo em que respeita restrições operacionais da rede.

A trajetória evolutiva das redes móveis ajuda a situar os desafios atuais. As redes 2G popularizaram a voz digital e permitiram o envio de mensagens curtas. O 3G viabilizou o acesso móvel à *internet* e introduziu as primeiras aplicações multimídia. O 4G consolidou o ecossistema de dados, suportando vídeo em alta definição e serviços em larga escala. O 5G introduziu paradigmas de baixa latência, alta confiabilidade e NS, abrindo espaço para aplicações de missão crítica. Por sua vez, o B5G/*Sixth Generation Mobile Networks* (6G) projeta um ambiente de hiperconectividade, com comunicações holográficas, *cell-free massive Multiple-Input and Multiple-Output (MIMO)* e redes orientadas por AI. Cada transição adicionou novas metas de desempenho e ampliou oportunidades para integrar algoritmos de aprendizado distribuído, reforçando a pertinência de investigar como o SL opera sob restrições reais de comunicação [3GPP TR 38.802 2017, Morocho-Cayamcela, Lee e Lim 2019].

A motivação prática deste trabalho decorre do impacto que falhas de comunicação exercem sobre aplicações sensíveis. Em saúde, sistemas de monitoramento remoto e diagnósticos assistidos por AI dependem de transmissão estável de dados biomédicos, interrupções ou atrasos podem comprometer a segurança do paciente. Em veículos autônomos, atrasos de poucos milissegundos podem levar a decisões equivocadas de frenagem ou desvio, com riscos de acidentes. Na indústria 4.0, descompassos de sincronização em células robóticas podem paralisar linhas de produção e gerar custos elevados. Esses casos compartilham a necessidade de redes resilientes capazes de assegurar latência mínima e confiabilidade elevada — atributos associados a URLLC — convivendo com demandas de eMBB e mMTC e exigindo gestão fina de *Quality of Service* (QoS) e de recursos rádio [Park et al. 2022, Khan et al. 2022]. De igual importância, a operação em borda (*edge*) impõe restrições energéticas, tornando crítico equilibrar acurácia de modelos, consumo e orçamento de comunicação para sustentar o aprendizado colaborativo em larga escala.

Apesar dos avanços, permanecem lacunas relevantes. Muitos estudos em aprendizado de máquina assumem redes ideais, ignorando variações realistas de latência, *jitter* e perda de pacotes; por sua vez, avaliações de redes móveis frequentemente desconsideram cargas reais de aprendizado distribuído. Poucos trabalhos conectam, de forma reprodutível, métricas de rede às métricas de aprendizado sob cenários controlados. Um ponto sensível é a escolha do ponto de corte (*cutlayer*) no SL, que divide o modelo entre cliente e servidor e introduz *trade-offs*: cortes rasos reduzem o custo computacional local, mas ampliam a dependência do enlace; cortes profundos aliviam o tráfego, porém sobrecarregam dispositivos limitados. A interação entre topologia de rede neural, dinâmica de tráfego e variabilidade da rede segue pouco explorada, e estudos recentes reforçam a necessidade de critérios sistemáticos para particionamento e otimização fim a fim [Matsubara, Levorato e Restuccia 2022, Dachille, Huang e Liu 2024]. Diferentemente de abordagens anteriores, este trabalho propõe um *framework* aberto, reprodutível e extensível, conectando explicitamente métricas de comunicação (latência, *jitter*, perdas, vazão e energia) a métricas de aprendizado (acurácia, estabilidade e convergência), de modo a permitir replicação, comparação justa e extensão por outros pesquisadores.

1.1 Pergunta de Pesquisa e Hipótese

A pergunta de pesquisa que orienta este trabalho pode ser formulada como:

> De que forma as métricas de rede — em especial latência, vazão, *jitter*, *Packet Loss Rate* (PLR) e consumo energético — afetam a acurácia e a convergência de modelos de SL em cenários B5G/6G?

A hipótese central é que a PLR exerce impacto mais crítico na convergência do que a latência isolada. Isso porque interrupções na transmissão comprometem diretamente o fluxo de treinamento. A latência, embora relevante, pode ser parcialmente compensada por ajustes na numerologia ou na largura de banda. Já as perdas de pacotes resultam em degradação imediata da qualidade do aprendizado. Assim, a expectativa é que a PLR se configure como métrica dominante, seguida por latência e vazão, com *jitter* e consumo energético em papéis secundários.

1.2 Justificativa

A justificativa para a realização deste estudo organiza-se em três dimensões interdependentes: científica, tecnológica e prática. Em conjunto, elas sustentam a relevância da investigação e orientam as escolhas metodológicas adotadas ao longo do trabalho.

No plano científico, esta dissertação busca avançar a compreensão das interações entre aprendizado distribuído e métricas de rede em ambientes móveis. Trata-se de um campo em consolidação, no qual ainda predominam abordagens voltadas à redução de latência ou à compressão de gradientes, com ganhos de eficiência, mas frequentemente dissociadas de cenários realistas de comunicação [Wu et al. 2023, Lin et al. 2024]. Estudos recentes demonstram a necessidade de conciliar eficiência comunicacional com preservação de privacidade e estabilidade do treinamento em arquiteturas sensíveis às condições de enlace [Ayad, Renner e Schmeink 2021, Zhang et al. 2023]. A originalidade deste trabalho reside em examinar, de modo integrado, como latência, *jitter*, perdas e vazão incidem sobre a acurácia e a convergência do SL em contextos móveis, aproximando a avaliação experimental de requisitos efetivamente encontrados no B5G/6G.

No plano tecnológico, a proposta ganha relevância ao oferecer um *framework* reprodutível. Esse ambiente integra o simulador *Network Simulator 3* (ns-3), o módulo Emulador de enlace para análise de redes *5G Link-Level Emulator for Network Analysis* (5G-LENA) e modelos de aprendizado em *Python*. A ênfase recai sobre rastreabilidade e replicação. A literatura já reconhece o ecossistema do ns-3 como base consolidada para estudos de redes, incluindo perspectivas sistemáticas sobre suas capacidades de validação [Yin et al. 2020, Campanile et al. 2020]. Ao articular essas ferramentas em um fluxo experimental claro, esta dissertação contribui para comparações consistentes entre cenários. Também facilita a reprodução por terceiros e amplia o escopo de testes sob diferentes perfis de tráfego, numerologias e configurações de enlace.

Quanto à escolha do paradigma colaborativo, optou-se pelo SL em detrimento do *Federated Learning* (FL). A decisão decorre da maior sensibilidade do SL às condições de rede: a cada iteração, ativações e gradientes atravessam o enlace, permitindo observar com granularidade como variações de latência, perdas e largura de banda influenciam

a estabilidade do treinamento. Essa propriedade atende diretamente ao objetivo de caracterizar o impacto das métricas de comunicação na qualidade do aprendizado, sobretudo quando se consideram dispositivos de borda e aplicações *Internet of Things* (IoT) com recursos computacionais limitados. Ao privilegiar a mensuração detalhada da interação rede–aprendizado, a abordagem adotada reforça a aderência entre desenho experimental e questões de pesquisa.

No eixo prático, a motivação decorre de demandas de setores críticos. Em tais contextos, indisponibilidade, atraso ou instabilidade comunicacional têm impacto imediato na segurança e na confiabilidade. No setor de saúde, por exemplo, o SL tem sido explorado justamente por permitir colaboração sem compartilhamento de dados brutos. Essa característica alinha requisitos de privacidade e desempenho em cenários sensíveis [Vepakomma et al. 2018, Shiranthika, Saeedi e Bajić 2023]. Em transporte e manufatura, atrasos de milissegundos podem degradar sistemas autônomos. Já interrupções ou sincronização deficiente comprometem cadeias de produção e aumentam custos operacionais. Esses cenários reforçam a necessidade de arquiteturas robustas no *Enhanced Data Rates for GSM Evolution* (EDGE), capazes de conciliar confiabilidade, eficiência energética e qualidade de serviço.

Sob essa perspectiva, o presente estudo combina rigor científico, inovação tecnológica e aderência a desafios reais. O objetivo central é avaliar, de forma integrada, os efeitos de latência, *jitter*, PLR, vazão e energia sobre a acurácia e a convergência do SL em cenários realistas B5G/6G. Ao alinhar desenho experimental, ferramentas reprodutíveis e aplicação prática, a dissertação pretende oferecer resultados sólidos, úteis à comunidade e coerentes com as exigências contemporâneas de redes e aprendizado distribuído.

1.3 Definição do problema

Esta pesquisa avança por duas direções complementares. A primeira é a implementação de suporte ao SL no ns-3, por meio do *ns-3 and AI Integration Framework* (ns3-ai), solução de código aberto amplamente utilizada e que viabiliza a integração entre o ciclo de treinamento distribuído e a pilha simulada de rede. A segunda direção investiga quais métricas de rede são mais críticas para o desempenho do SL, com foco em latência, vazão, *jitter*, taxa de perda de pacotes (PLR) e energia. Para garantir interpretações causais mais claras, comparamos dois cenários: (i) rede com carga de SL (ativações/gradientes trafegando a cada iteração) e (ii) rede isolada (mesma configuração, mas sem o fluxo de SL), usada como linha de base.

No SL *Vanilla*, o treinamento entre servidor e clientes é sequencial: cada cliente precisa se comunicar com o servidor a cada iteração do conjunto de treinamento [Ryu, Won e Lee 2022, Wu et al. 2023, Lin et al. 2024]. Essa simplicidade metodológica

é útil, mas vem acompanhada de uma consequência prática: a falha de um único enlace pode bloquear o ciclo inteiro, elevando a sensibilidade do sistema a condições adversas de rede [Wu et al. 2023, Oh et al. 2022]. Assim, compreender quais métricas efetivamente limitam o SL não é apenas um detalhe de desempenho; é uma condição para viabilizar a sua adoção em cenários reais.

A literatura aponta a latência como uma das métricas mais determinantes para o SL, sobretudo quando há muitos dispositivos participantes ou quando os equipamentos são limitados em recursos computacionais [Lin et al. 2024, Wu et al. 2023]. Neste trabalho, tratamos a latência como hipótese principal e, ao mesmo tempo, ampliamos o escopo para quantificar o papel relativo de vazão, *jitter*, PLR e energia. Em outras palavras, perguntamos: o que, de fato, mais impede o SL de convergir com qualidade e em tempo hábil? Para responder, adotamos o SL *Vanilla* [Ryu, Won e Lee 2022] na experimentação, visando contrastamos o comportamento do sistema e seu desempenho final em relação as métricas da rede móvel *Beyond Fifth or Sixth Generation Wireless Networks* (B5G/6G).

Mesmo sem interrupções totais, condições instáveis — baixa largura de banda, latência elevada, *jitter* ou perdas — degradam a comunicação, induzindo atrasos, flutuações no passo de treinamento e, em casos extremos, falhas [Wu et al. 2023, Oh et al. 2022]. Embora existam propostas de mitigação (p.ex., paralelismo parcial, compressão de ativações ou reorganização do pipeline), o objetivo central é medir e comparar o impacto das métricas de rede sobre o SL, ressaltando qual métrica emerge como gargalo predominante quando contrastamos SL versus rede isolada.

Existem ainda fatores que podem interromper ou dificultar o treinamento e que, embora não componham os experimentos desta dissertação, ajudam a situar o problema: (i) desconexões de rede, que fazem todos aguardarem a reconexão de um cliente e ampliam o tempo total [Wu et al. 2023]; (ii) interrupções no dispositivo, como desligamento por bateria ou término abrupto da aplicação [Wu et al. 2023]; e (iii) limitações computacionais locais, especialmente quando a camada de corte se encontra mais profundamente na rede neural [Lin et al. 2024, Wu et al. 2023].

Em contextos práticos, projetar sistemas de SL significa equilibrar capacidade computacional e condições de rede. Mecanismos como *timeouts*, tolerância a falhas e otimizações de uso de recursos aumentam a robustez. No presente estudo, o cenário é idealizado e controlado, o que facilita o isolamento dos efeitos. Ainda assim, a comparação com a rede isolada permanece peça-chave: ela separa o impacto intrínseco da rede daquele induzido pela própria carga do SL, permitindo identificar, com maior clareza, quais métricas realmente importam e quanto elas importam.

1.4 Objetivos

1.4.1 Objetivo Geral

O objetivo geral desta dissertação é propor e avaliar uma arquitetura de integração entre o simulador ns-3, 5G-LENA e o SL, investigando o impacto das condições de rede sobre o desempenho do treinamento distribuído. Pretende-se analisar como métricas de comunicação (latência, vazão, *jitter*, perdas e consumo energético) influenciam métricas de aprendizado (acurácia, tempo de treinamento e robustez do modelo), bem como explorar soluções que mitiguem limitações observadas.

1.4.2 Objetivos Específicos

Com base no objetivo geral, foram definidos os seguintes objetivos específicos:

- Estudar os fundamentos de redes móveis B5G/6G e sua interação com paradigmas de aprendizado distribuído;
- Identificar métricas relevantes de rede (latência, vazão, *jitter*, perdas e energia) e de desempenho do modelo (acurácia, tempo de treino);
- Propor e implementar uma arquitetura de integração entre o ns-3 e SL;
- Realizar simulações variando cenários de rede e de treinamento, avaliando seu impacto sobre a comunicação e sobre o modelo de *Machine Learning* (ML);
- Investigar a influência de perfis de tráfego (URLLC, eMBB, mMTC), numerologias e IoT no desempenho conjunto da rede e do SL;
- Avaliar a escalabilidade e a robustez do SL em ambientes heterogêneos de B5G.

1.5 Contribuições

As principais contribuições desta dissertação podem ser organizadas em cinco eixos complementares, que articulam aspectos de integração, algoritmo, experimentação e análise conjunta de redes e aprendizado distribuído:

a) Arquitetura de Integração: foi concebida e validada a arquitetura *SplitLearning-ns3* [LABORA-INF/UFG 2025], que integra de forma inédita o modelo distribuído de SL com o simulador de redes ns-3, utilizando o módulo 5G-LENA e a interface ns3-ai para comunicação eficiente entre os ambientes de simulação de rede (*C++*) e de aprendizado de máquina (*Python*).
b) Projeto Algorítmico: foi implementado um algoritmo de integração entre cliente, servidor e a infraestrutura de rede simulada, capaz de coordenar a propagação direta

e a retropropagação entre os dispositivos clientes e o servidor, considerando atrasos, perdas de pacotes e restrições reais da rede.

c) SL em Redes B5G: foi desenvolvida e avaliada uma aplicação de SL baseada em uma *Convolutional Neural Network* (CNN) com particionamento entre cliente e servidor, utilizando o conjunto de dados *Modified National Institute of Standards and Technology* (MNIST). A proposta permitiu examinar o impacto da divisão da rede neural na eficiência computacional e no desempenho de aprendizagem em dispositivos com recursos limitados.

d) Avaliação Experimental com ns-3 5G-LENA: foram conduzidos experimentos em cenários que incluem múltiplos perfis de tráfego (URLLC, eMBB e mMTC), variações de numerologia e fatiamento por IoT, analisando seus impactos sobre métricas de rede como latência, vazão, perda de pacotes e consumo energético.

e) Métricas de Desempenho Conjunto: foi estabelecida uma análise integrada da variabilidade da acurácia do modelo em função da distância dos dispositivos ao *Next Generation NodeB*s (gNBs), do perfil de tráfego e das condições de rede, demonstrando de forma inédita a correlação entre métricas de comunicação (latência, vazão, perdas, energia) e métricas de aprendizado (acurácia e tempo de treinamento).

Essas contribuições evidenciam que a proposta vai além da caracterização de redes móveis, consolidando uma abordagem metodológica para investigar, de forma unificada, os efeitos de condições de rede sobre o desempenho de modelos de aprendizado distribuído. O diferencial está na documentação e validação da integração entre sistemas de simulação de rede e algoritmos de SL, destacando os desafios impostos pela latência e pela heterogeneidade de tráfego em cenários B5G.

1.5.1 Organização da Proposta

Com a intenção de apresentar de forma clara os conceitos, a fundamentação teórica, a metodologia empregada e os resultados obtidos, esta dissertação foi estruturada conforme a seguinte organização:

- Capítulo 1 – Introdução: apresenta o contexto em que se insere esta pesquisa, destacando a evolução das redes móveis, a relevância da AI distribuída e a motivação para o uso do SL em cenários B5G/6G. Também discute a justificativa científica, tecnológica e prática do estudo, formula a pergunta de pesquisa, explicita os objetivos gerais e específicos e sintetiza as principais contribuições da dissertação;
- Capítulo 2 – Conceitos e Trabalhos Relacionados: reúne os conceitos fundamentais que sustentam o desenvolvimento da proposta, incluindo a descrição do paradigma de SL, o integrador ns3-ai, o módulo 5G-LENA, o simulador ns-3 e as características

das redes móveis 5G/B5G. De modo complementar, revisa criticamente os trabalhos relacionados, organizados em três eixos: avanços no aprendizado de máquina, mecanismos em redes móveis e a interface entre esses dois domínios. A seção de síntese aponta as lacunas existentes e como esta pesquisa busca preenchê-las;

- Capítulo 3 – SL como habilitador para Redes Sem Fio e Móveis: descreve a arquitetura proposta, detalhando o cenário de sistema, a estrutura do modelo SL, o fluxo de execução em etapas, a reprodutibilidade dos experimentos e a integração com o modelo 5G. São apresentados os mecanismos de configuração de canais, fatiamento por numerologias e *Bandwidth Parts* (BWPs), além da modelagem de tráfego *On–Off* e da arquitetura integrada cliente–servidor. Este capítulo constitui o núcleo da proposta metodológica;
- Capítulo 4 – Avaliação e Resultados: apresenta a metodologia experimental e os resultados obtidos. A análise é organizada por métricas de rede (latência, vazão, *jitter*, PLR e consumo de energia) e por *slices*/BWPs. São discutidos os impactos sobre a acurácia e a convergência do SL, sempre sob a lógica causa–efeito. O capítulo inclui tabelas de síntese e gráficos de dispersão, permitindo identificar os fatores mais críticos para o desempenho do aprendizado distribuído;
- Capítulo 5 – Conclusão e Trabalhos Futuros: reúne as principais conclusões, costurando os achados experimentais às hipóteses levantadas na introdução. Além de reforçar as contribuições científicas, tecnológicas e práticas, são apresentadas perspectivas de continuidade, incluindo comparações entre SL e FL, uso de *datasets* mais complexos (como *Canadian Institute For Advanced Research — 10 classes* (CIFAR-10)), avaliação de cenários multi-gNB e validação em *hardware* real.

Essa organização tem como objetivo oferecer uma narrativa progressiva e coesa, que parte da contextualização do problema, passa pela fundamentação e pela metodologia, e culmina com a análise crítica dos resultados e a projeção de novas pesquisas na interseção entre SL e redes móveis B5G/6G.

CAPÍTULO 2

Conceitos e Trabalhos Relacionados

Este capítulo apresenta os principais conceitos associados ao SL e analisa criticamente a literatura correlata. O objetivo é situar o SL no contexto do aprendizado colaborativo, descrever o integrador ns3-ai e avaliar como pesquisas anteriores trataram da integração entre redes móveis e algoritmos de aprendizado distribuído.

2.1 Conceitos Fundamentais

2.1.1 Split Learning

O SL é um paradigma de aprendizado colaborativo no qual a rede neural é dividida entre cliente e servidor. O cliente treina as camadas iniciais e envia ativações intermediárias ao servidor, que completa o processo. Essa divisão reduz a carga computacional local e preserva os dados brutos no dispositivo, mas cria dependência direta das condições de rede, especialmente em termos de latência, *jitter* e PLR [Chen, Li e Chakrabarti 2021, Ryu, Won e Lee 2022].

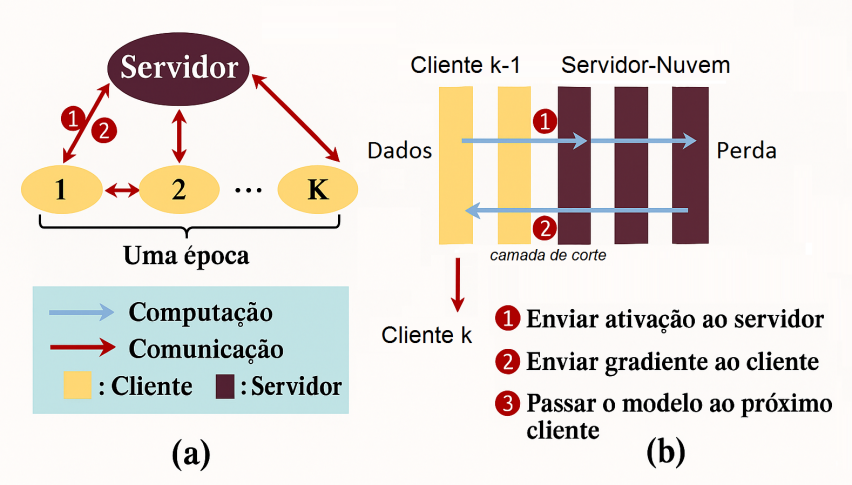

Figura 2.1: Fluxo de treinamento no *Split Learning Vanilla*: (a) ciclo por época com comunicação cliente–servidor; (b) divisão do modelo em *cutlayer*, com envio de ativações ao servidor e retorno de gradientes ao cliente [Chen, Li e Chakrabarti 2021].

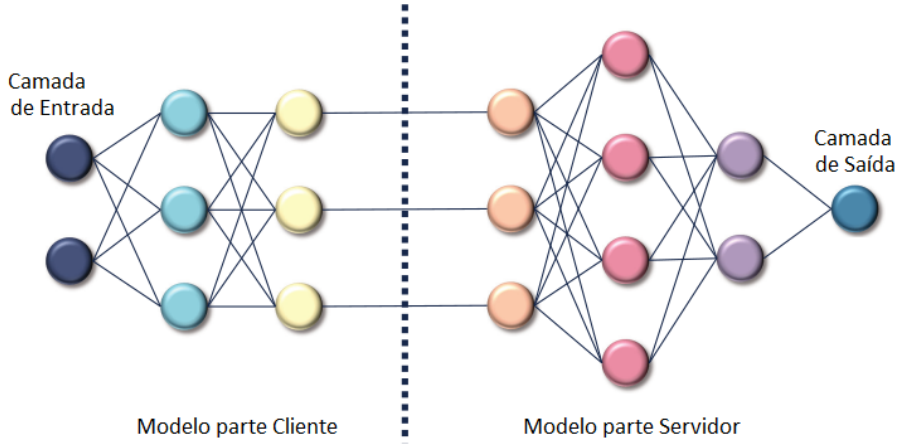

Figura 2.2: Divisão de uma rede neural em modelo do cliente (camadas iniciais) e modelo do servidor (camadas finais), representando o princípio central do SL [Ryu, Won e Lee 2022].

A Figura 2.1 mostra que o SL opera em um ciclo: (i) o cliente envia ativações ao servidor, o que reduz sua carga local, mas aumenta a dependência da rede; (ii) o servidor processa gradientes e os devolve, permitindo a atualização local, mas introduzindo atraso sempre que houver latência ou perdas; (iii) o modelo é repassado para o próximo cliente, garantindo continuidade, porém ampliando o risco de propagação de instabilidades. Na Figura 2.2 evidencia a existência do ponto de corte na rede neural. Em termos práticos, escolher onde cortar implica um equilíbrio: cortes mais precoces tendem a aliviar o processamento no cliente, ao custo de maior tráfego; já cortes mais profundos reduzem as comunicações, mas podem sobrecarregar dispositivos com recursos limitados. Essa relação de troca torna o SL particularmente sensível às condições de rede e à capacidade do dispositivo — um aspecto que orienta as decisões ao longo desta dissertação.

Com base nessa dualidade, a Tabela 2.1 sintetiza as diferenças fundamentais entre FL e SL, destacando os efeitos práticos de cada paradigma.

Tabela 2.1: Comparação entre FL e SL

Aspecto	**FL**	**SL**
Compartilhamento	Gradientes/pesos atualizados	Ativações intermediárias
Custo computacional local	Elevado em dispositivos limitados	Reduzido, adequado para IoT
Dependência de rede	Menor, apenas para sincronização periódica	Maior, pois há envio de ativações a cada iteração
Divisão do modelo	Cada cliente treina o modelo completo localmente	Modelo dividido em camadas cliente–servidor
Privacidade	Dados locais preservados, mas vulneráveis a inferência de gradientes	Dados brutos nunca saem do cliente

Além do SL tradicional, surgiram variantes como *Parallel Split Learning* (PSL), *Split Federated Learning* (SFL) e *Efficient Parallel Split Learning* (EPSL). Essas variantes refletem diferentes estratégias: 1) aliviar o cliente pode acelerar o treinamento, mas aumenta o tráfego; 2) por outro lado, reduzir comunicações pode preservar rede, mas impactar a acurácia. A Tabela 2.2 apresenta um resumo dessas variações.

Tabela 2.2: Variações do SL

Variante	**Característica Principal**	**Vantagem**	**Limitação**
EPSL	Compressão de ativações/-gradientes	Menor tráfego de rede	Possível perda de acurácia
PSL	Treinamento paralelo de múltiplos clientes	Acelera o processo global	Elevada demanda de rede síncrona
SFL	Combinação de FL e SL	Reduz frequência de comunicação	Complexidade de coordenação

2.1.2 Sensibilidade do SL às Métricas de Rede

O desempenho do SL não depende apenas da divisão de camadas, mas também da qualidade do enlace de comunicação. Métricas clássicas de rede como atraso (latência), vazão (*throughput*), *jitter* e PLR impactam diretamente a estabilidade temporal do processo de treinamento, o tempo de convergência e a acurácia do modelo. A Tabela 2.3 resume, em alto nível, a relação entre essas métricas e o SL.

Tabela 2.3: Relação conceitual - como cada métrica de rede afeta o SL

Métrica	**Tendência ↑**	**Impacto esperado**	**Acurácia**
Energia (J)	↑	Maior custo de execução	neutro
Jitter (s)	↑	Instabilidade na sincronização	↓
Latência (s)	↑	Sincronização mais lenta	↓
PLR (%)	↑	Perda de ativações/gradientes	↓
Vazão (Mbps)	↑	Maior troca de ativações/gradientes	↑

Latência. Quanto mais elevadas por iteração aumentam o tempo de convergência, pois cada ciclo cliente–servidor (envio de ativações z e retorno de gradientes g) sofre atrasos acumulados. Contudo, desde que a confiabilidade do enlace se mantenha (PLR baixa), a acurácia final pode ser preservada. A Figura 2.3 ilustra essa expectativa conceitual, que será validada empiricamente na seção 4.1.1.1.

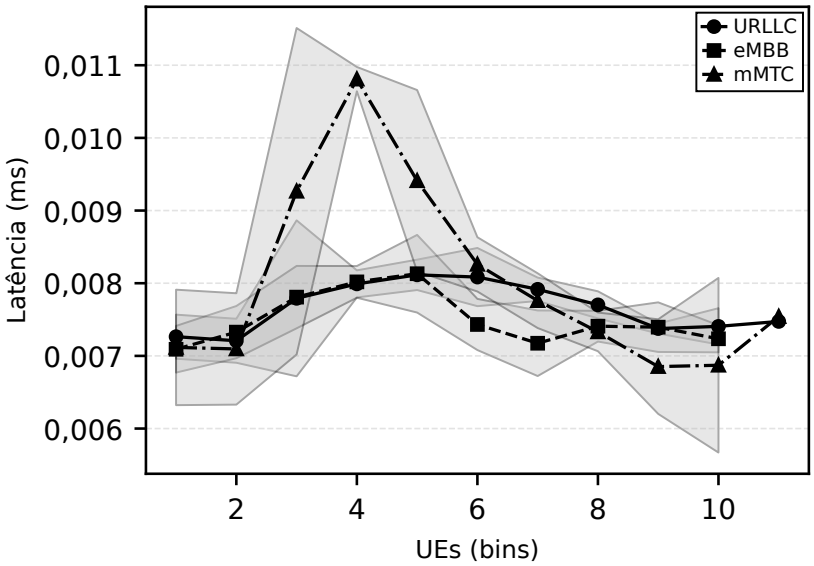

Figura 2.3: Visão teórica da latência no SL - aumentos de *delay* por iteração tendem a alongar o tempo de convergência; mantendo-se a confiabilidade PLR baixa, a acurácia final pode ser preservada.

Essa representação deixa claro que a latência não deve ser interpretada apenas como um simples atraso, mas como um fator que redefine a escalabilidade do SL em cenários densos. Em aplicações sensíveis, como saúde e veículos autônomos, mesmo pequenas dilatações do tempo de convergência podem comprometer a viabilidade prática.

Vazão. A disponibilidade de maior vazão de rede aumenta a capacidade de transmissão das ativações e dos gradientes entre cliente e servidor, reduzindo o tempo total necessário para completar o ciclo de treinamento. O efeito esperado é, portanto, uma aceleração do processo de convergência. No entanto, o impacto sobre a acurácia final tende

a ser limitado, já que a confiabilidade do enlace — refletida pela PLR — exerce papel mais decisivo na qualidade do aprendizado. A Figura 2.4 ilustra esse comportamento esperado, servindo como hipótese conceitual a ser confrontada com os resultados quantitativos na seção 4.1.1.2. Da mesma forma, evidencia que priorizar somente a vazão pode gerar ganhos de velocidade, mas não necessariamente de qualidade no aprendizado. Isso reforça a necessidade de políticas de escalonamento que equilibrem vazão com confiabilidade do enlace.

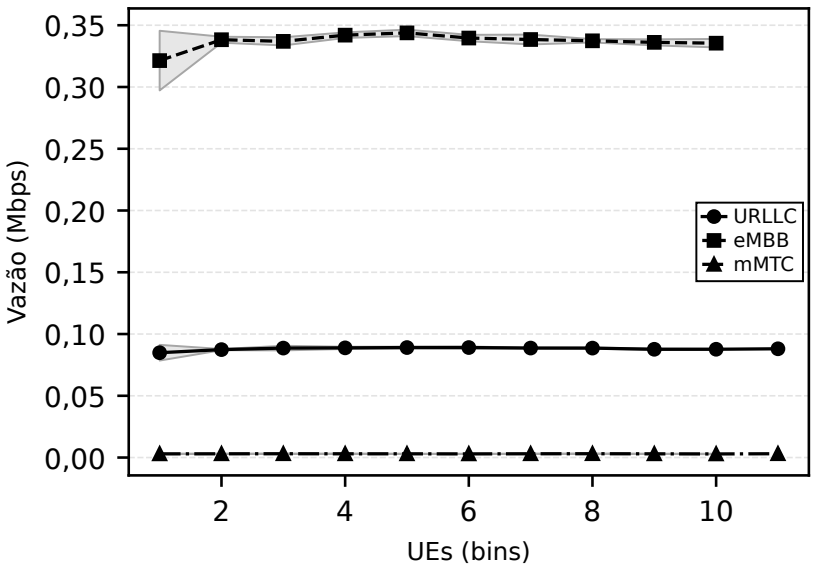

Figura 2.4: Visão teórica do *throughput* no SL - maior vazão reduz o tempo de treinamento, mas o impacto direto na acurácia final permanece limitado.

Jitter. Entendido como a variabilidade da latência entre diferentes iterações, é um fator crítico para a estabilidade do processo de treinamento no SL. Mesmo que os valores médios de atraso se mantenham baixos, a presença de flutuações significativas pode introduzir instabilidades na curva de aprendizado, tornando a convergência menos previsível. Esse comportamento é especialmente sensível em cenários de tráfego intermitente, como no *slice* mMTC, onde o padrão *On–Off* acentua a irregularidade temporal. A Figura 2.5 sintetiza esse efeito esperado, servindo como hipótese a ser confrontada com os resultados empíricos na seção 4.1.1.3.

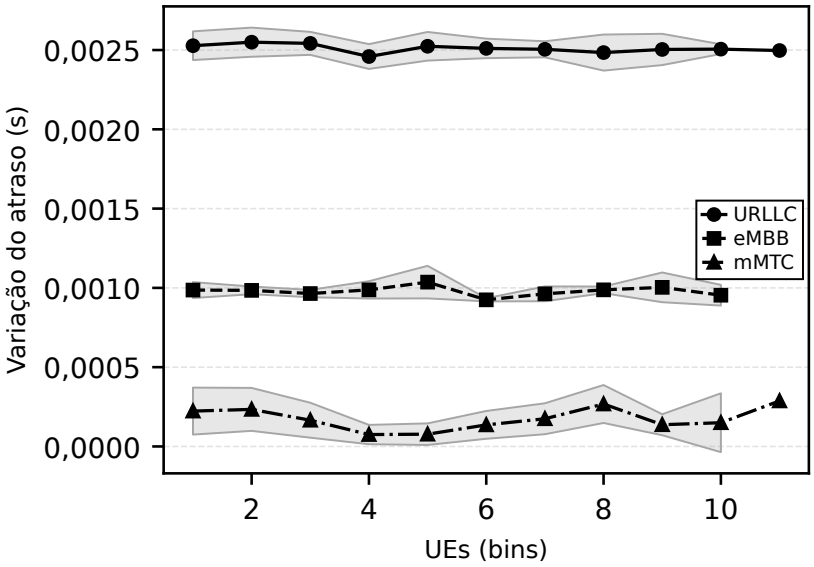

Figura 2.5: Visão teórica do *jitter* no SL - flutuações na latência comprometem a estabilidade da curva de aprendizado, mesmo quando a acurácia média final se mantém.

Esse resultado sugere que, em ambientes móveis, o *jitter* pode ser tão prejudicial quanto a própria latência média. Sua presença exige estratégias adicionais de compensação temporal, sem as quais a convergência do modelo se torna errática.

PLR. Representa a fração de ativações ou gradientes perdidos durante a transmissão. Diferente da latência ou do *jitter*, que afetam principalmente o tempo de convergência e a suavidade da curva de aprendizado, a PLR exerce efeito direto sobre a acurácia final: a perda de gradientes implica redução efetiva da informação de treinamento, comprometendo o processo de atualização dos parâmetros do modelo. A Figura 2.6 sintetiza essa expectativa conceitual, na qual valores elevados de PLR resultam em quedas acentuadas de acurácia, servindo como hipótese a ser validada empiricamente na seção 4.1.1.4.

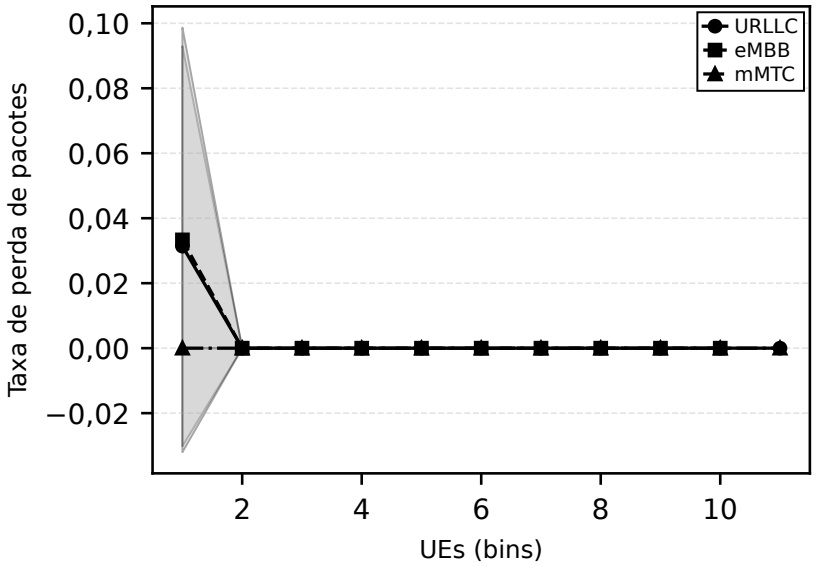

Figura 2.6: Visão teórica da PLR no SL - aumentos na perda de pacotes provocam queda direta na acurácia de validação.

Essa Figura 2.6 reforça a hipótese central da dissertação: entre todas as métricas, a PLR é a mais crítica, pois causa perdas irreversíveis de informação. Em cenários de alta variabilidade, pequenas taxas de perda já comprometem a robustez do SL.

Energia. O consumo de energia está associado principalmente ao custo de execução do treinamento e à autonomia dos dispositivos, mas não exerce impacto direto sobre a acurácia final. Cenários de maior gasto energético implicam em maior custo de operação e podem limitar a escalabilidade do SL em dispositivos restritos, como IoT. A Figura 2.7 resume esse efeito esperado, que será detalhado empiricamente na seção 4.1.1.5.

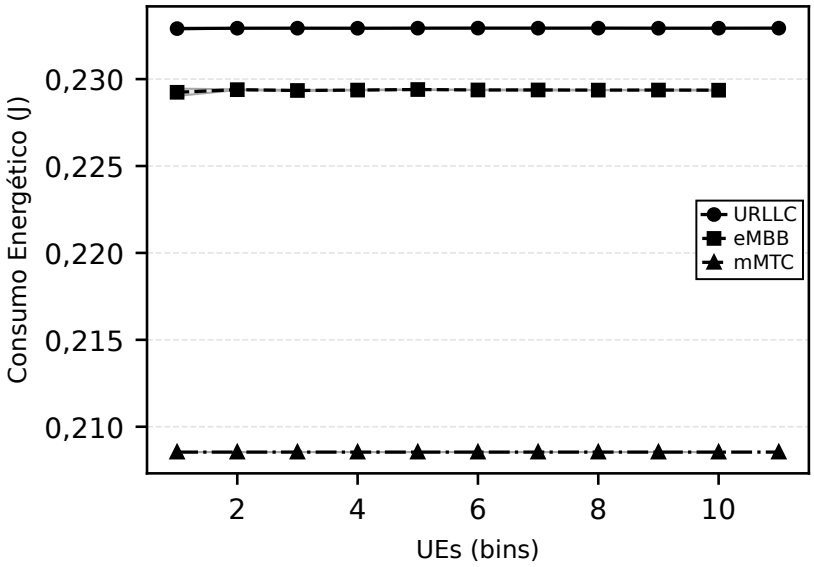

Figura 2.7: Visão teórica do consumo de energia no SL - maiores gastos aumentam o custo de execução, mas não afetam diretamente a acurácia final do modelo.

Embora a acurácia não seja afetada diretamente, o consumo energético atua como barreira prática para adoção em larga escala. Modelos teoricamente eficientes podem se tornar inviáveis em cenários de IoT e dispositivos de baixa potência.

Resumo. As figuras conceituais de latência, vazão, *jitter*, PLR e energia (Figuras 2.3, 2.4, 2.5, 2.6 e 2.7) complementam a Tabela 2.3, oferecendo uma visão preliminar dos efeitos esperados das métricas de rede sobre o SL. Esses modelos teóricos orientam a análise experimental do Capítulo 4 (subseções 4.1.1.1, 4.1.1.2, 4.1.1.3, 4.1.1.4 e 4.1.1.5) e se articulam com a formalização e os procedimentos metodológicos definidos no Capítulo 3.

Ponte com os resultados experimentais. A presente síntese conceitual orienta a interpretação dos resultados empíricos no Capítulo 4, no qual tais métricas são quantificadas em cenários 5G realistas. Em particular, a relação entre latência média por iteração e acurácia de validação é apresentada na seção 4.1.1.1, por meio da Figura 4.1, que confirma a tendência antecipada na Figura 2.3 e detalha as faixas observadas de atrasos e acurácia nos diferentes cenários simulados.

2.1.3 Integrador ns3-ai

O ns3-ai é um módulo complementar ao simulador ns-3 que viabiliza a integração entre algoritmos de aprendizado de máquina (como *PyTorch* e *TensorFlow*) e a simulação de redes de comunicação. A comunicação entre os domínios *C++* e *Python* é realizada por meio de memória compartilhada (*shared memory*), o que reduz significativamente

a sobrecarga de troca de mensagens e garante maior desempenho em comparação a abordagens baseadas em *sockets* [Yin et al. 2020, Nakashima et al. 2022].

Essa arquitetura permite que modelos de AI sejam treinados ou ajustados dinamicamente durante a execução da simulação, viabilizando cenários de aprendizado *online*. Essa característica é fundamental para estudos de sistemas adaptativos, em que decisões de controle dependem de dados gerados em tempo real, tornando o ns3-ai uma ferramenta especialmente relevante para pesquisas em redes inteligentes e adaptativas.

O módulo apresenta desempenho superior a alternativas como o `ns3-gym`, alcançando velocidades de transferência até 50 a 100 vezes maiores em experimentos de grande volume de dados. Essa eficiência torna-o particularmente adequado para aplicações que exigem elevada interação entre o simulador e o modelo de aprendizado, como em treinamentos de redes neurais profundas em tempo real.

Além disso, o ns3-ai fornece uma interface de alto nível com suporte a diferentes estruturas de aprendizado, incluindo aprendizado por reforço (*Reinforcement Learning* (RL)) e aprendizado profundo (*Deep Learning* (DL)), bem como tipos de dados definidos pelo usuário. O mecanismo de sincronização simples garante a execução sequencial dos processos, favorecendo a reprodutibilidade dos experimentos, aspecto crucial em trabalhos acadêmicos.

Diversos casos de uso já foram relatados na literatura: a própria estrutura do ns3-ai foi apresentada e validada para integrar algoritmos de ML a experimentos de rede em [Yin et al. 2020]; aplicações incluem controle de taxa em WLAN com *Deep Reinforcement Learning* (DRL) [Nakashima et al. 2022], a simulação de cenários de FL com o ns-3 [Ekaireb et al. 2022] e estudos de roteamento/otimização em redes subaquáticas com ns3-ai [Shruthi 2024]. Há ainda trabalhos que exploram a predição de métricas de enlace por aprendizado profundo — por exemplo, potência recebida e medidas correlatas úteis à estimação de *Channel Quality Indicator* (CQI) [Koda et al. 2020, Yu et al. 2024] —, bem como avaliações de escalonadores e QoS em 5G *5G New Radio* (5G NR)/5G-LENA [Koutlia et al. 2023, Lagén et al. 2023]. Em todos esses cenários, o ns3-ai tem se mostrado uma solução robusta e eficiente para a integração entre redes simuladas e algoritmos de aprendizado de máquina.

Dessa forma, o ns3-ai se consolida como um integrador estratégico para o desenvolvimento de soluções de próxima geração em redes móveis, unindo realismo de simulação, flexibilidade metodológica e alto desempenho computacional. Essas características o tornam essencial para pesquisas que buscam compreender a interação entre parâmetros de rede e modelos de aprendizado distribuído.

2.1.4 5G-LENA

O LTE-NR, na versão 5G-LENA, constitui o módulo do simulador ns-3 dedicado à modelagem e avaliação de redes móveis de quinta geração. Desenvolvido pelo grupo *Centre Tecnològic de Telecomunicacions de Catalunya* (CTTC), o 5G-LENA viabiliza experimentos reprodutíveis que contemplam aspectos de acesso rádio, alocação de recursos e mecanismos de *slicing* em conformidade com as especificações do *Third Generation Partnership Project* (3GPP).

Em sua arquitetura, o 5G-LENA incorpora funcionalidades como múltiplas numerologias, BWPs, modos *duplex* (*Frequency Division Duplex* (FDD)/*Time Division Duplex* (TDD)) e suporte a diferentes tipos de tráfego (URLLC, eMBB, mMTC). Essa flexibilidade possibilita a criação de cenários heterogêneos, nos quais requisitos de latência, confiabilidade e eficiência espectral coexistem. Outro ponto de destaque é o suporte à instrumentação detalhada, permitindo extrair métricas de desempenho como atraso fim a fim, vazão, taxa de perda de pacotes, consumo energético e eficiência espectral.

No contexto desta dissertação, o 5G-LENA é peça-chave para aproximar a simulação dos desafios reais enfrentados em redes móveis além da quinta geração (B5G/6G). Sua integração com o ns3-ai possibilita vincular as métricas de rede às métricas de aprendizado em SL, criando um ambiente unificado em que algoritmos de AI são avaliados sob condições realistas de comunicação. Dessa forma, o 5G-LENA não é apenas um módulo de rede, mas um facilitador científico que garante fidelidade, transparência e relevância prática aos experimentos conduzidos.

2.1.5 Simulador de rede ns-3

O ns-3 é um simulador de código aberto amplamente reconhecido pela comunidade acadêmica e industrial para a avaliação de redes de comunicação. Seu desenvolvimento colaborativo garante aderência contínua a padrões internacionais, como os definidos pelo *Institute of Electrical and Electronics Engineers* (IEEE) e pelo 3GPP. Ao contrário de *testbeds* físicos, o ns-3 oferece um ambiente controlado e reprodutível, no qual é possível isolar variáveis e avaliar sistematicamente protocolos, arquiteturas e algoritmos de comunicação.

A organização do ns-3 fundamenta-se na distinção entre módulos e modelos. Os módulos correspondem a bibliotecas vinculáveis (como *core*, *network*, *internet*, *wifi*, *nr*), enquanto os modelos representam abstrações de protocolos e dispositivos específicos. Essa separação metodológica garante maior transparência na configuração dos cenários e permite carregar apenas os componentes necessários, favorecendo a reprodutibilidade científica. A Figura 2.8 ilustra essa estrutura em camadas, destacando como a modularidade do simulador apoia a integração de diferentes componentes em um ambiente unificado.

Além disso, a base em *C++* com exportação quase integral da *API* para *Python* cria o ambiente propício à integração com ferramentas externas, como a interface ns3-ai, utilizada neste trabalho para orquestrar experimentos de SL em tempo de simulação. Outro aspecto relevante é o ecossistema de documentação oficial do ns-3, estruturado em pilares complementares: *Tutorial*, *Manual*, *Model Library*, *Doxygen* e *Contributing*. Esses guias versionados oferecem suporte técnico consistente e legitimam as decisões metodológicas deste trabalho, assegurando que a configuração dos cenários permaneça aderente às práticas aceitas pela comunidade de pesquisa.

Neste trabalho, o ns-3 atua como núcleo de experimentação. Sua capacidade de gerar métricas detalhadas — incluindo latência, vazão, *jitter*, perda de pacotes e consumo energético — permite estabelecer uma relação direta entre condições de rede e desempenho de modelos de aprendizado colaborativo. Ao integrá-lo ao módulo 5G-LENA e à interface ns3-ai, constrói-se uma infraestrutura aberta e reprodutível, capaz de aproximar a simulação de cenários das redes móveis reais e de apoiar investigações sobre SL em ambientes B5G/6G.

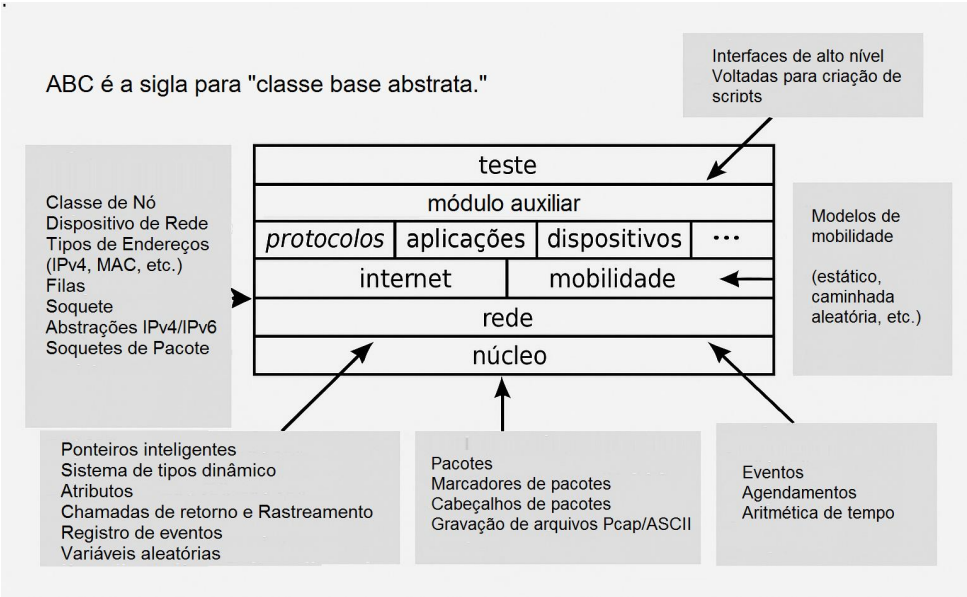

Figura 2.8: Estrutura de *software* do ns-3: a separação em camadas e a distinção entre módulos e modelos facilitam a composição de cenários e ampliam a reprodutibilidade dos experimentos [ns-3 Project 2025].

2.1.6 Redes móveis 5G/B5G

A quinta geração de redes móveis (5G) consolida um marco tecnológico ao introduzir *network slicing*, múltiplas numerologias e a oferta de serviços heterogêneos sobre a mesma infraestrutura. Na perspectiva de engenharia de redes, o 5G foi concebido para disponibilizar velocidades na ordem de gigabits, latências de poucos milissegundos

e capacidade ampliada para suportar densidades elevadas de dispositivos, estabelecendo a base para cenários B5G/6G e para aplicações sensíveis a tempo e confiabilidade [Kurose e Ross 2020]. Em conformidade com as especificações do 3GPP [3GPP 2019], essa arquitetura é sustentada por três pilares fundamentais de tráfego: URLLC, voltado a aplicações críticas que exigem latência extremamente baixa e alta confiabilidade; eMBB, destinado a serviços de alta demanda de dados (por exemplo, vídeo 4K/8K e realidade aumentada); e mMTC, que suporta a densidade massiva de dispositivos de IoT. A Figura 2.9 ilustra essa rede celular global como uma *network of networks*, destacando a complexidade da infraestrutura que sustenta tais serviços. Essa caracterização posiciona o 5G como ponto de inflexão para investigar a viabilidade do SL em redes móveis de próxima geração.

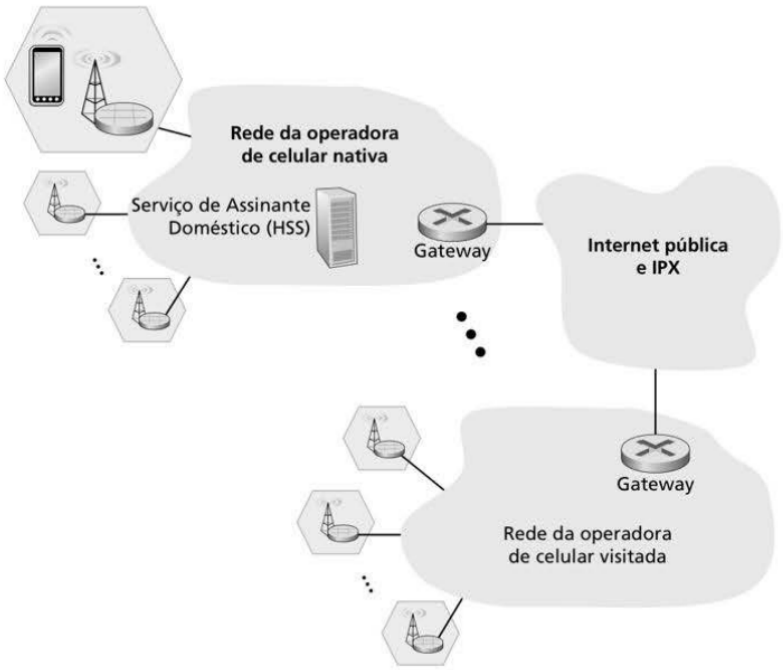

Figura 2.9: Rede celular global como *network of networks* [Kurose e Ross 2020].

Enquanto o *Fourth Generation Mobile Networks (IMT-Advanced)* (4G) teve como prioridade a ampliação da banda larga móvel, o 5G expande o escopo para cenários de missão crítica, incluindo veículos autônomos, telemedicina e Industria 4.0. O efeito imediato dessa diversificação é o aumento da complexidade no gerenciamento de recursos, que demanda novas estratégias de orquestração da rede e abre espaço para a integração de técnicas de aprendizado de máquina [Koutlia et al. 2023].

O termo B5G e, em seguida, o 6G projetam cenários ainda mais ambiciosos, marcados por conceitos como hiperconectividade, comunicações holográficas, *cell-free massive MIMO* e redes auto-organizáveis orientadas por AI. Esses avanços respondem à necessidade de atender requisitos heterogêneos de tráfego em ambientes dinâmicos e imprevisíveis.

Nesse contexto, a avaliação do SL em cenários 5G/B5G representa uma oportunidade científica relevante. A diversidade de serviços e suas demandas específicas impõem variações de atraso, confiabilidade de enlace e eficiência energética, que funcionam como barreiras práticas à adoção de algoritmos colaborativos. Assim, ao investigar a resiliência do SL frente a esses fatores, esta dissertação não apenas contribui para o estado da arte da pesquisa em aprendizado colaborativo em redes móveis, como também projeta sua aplicabilidade em setores críticos da sociedade.

2.1.7 Fatiamento de Rede (NS)

O fatiamento de rede (NS) no 5G NR permite particionar logicamente a mesma infraestrutura física em fatias (*slices*) independentes, cada qual com requisitos próprios de QoS (p.ex., URLLC, eMBB, mMTC). Em termos práticos, cada fatia recebe um perfil de recursos e políticas sob medida para a sua classe de serviço, o que viabiliza flexibilidade e especialização na entrega. Segundo as especificações 3GPP ([3GPP TS 23.501 2025, 3GPP TS 23.502 2025]) e o *white paper* do [NGMN Alliance 2015], o controle ponta-a-ponta no núcleo 5G é exercido por funções como *Access and Mobility Management Function* (AMF), *Session Management Function* (SMF) e *Policy Control Function* (PCF), que asseguram isolamento lógico e conformidade com metas de QoS; ainda assim, o isolamento no domínio *Radio Access Network* (RAN) é parcial, dado o compartilhamento de espectro e demais recursos de acesso [3GPP TS 23.501 2025, NGMN Alliance 2015, Popovski et al. 2018, Ebrahimi et al. 2024].

No domínio rádio, o NS é materializado por perfis de recursos e parâmetros específicos (largura útil e numerologia) instanciados via BWPs e selecionados/comutados por controle *Radio Resource Control* (RRC). Cada BWP corresponde a um bloco contíguo de recursos em frequência configurado com uma numerologia específica, na qual o espaçamento de subportadora é dado por $\Delta f(\mu) = 15 \cdot 2^{\mu}$ kHz e a duração do *slot* por $T_{\text{slot}} = \frac{1\,\text{ms}}{2^{\mu}}$, tipicamente com $\mu \in \{0,1,2,3\}$. Essa seleção dinâmica de BWPs permite alinhar o *slice* ao perfil de tráfego: URLLC tende a adotar numerologias mais altas (menor T_{slot}) para reduzir latência e elevar confiabilidade, enquanto eMBB privilegia maior largura útil para maximizar *throughput*. Tais escolhas impactam diretamente métricas como latência, *jitter* e taxa de perda de pacotes (PLR), exigindo coordenação eficiente dos recursos compartilhados na RAN [3GPP TS 38.211 2024, 3GPP TS 38.213 2022, 3GPP TS 38.214 2025, 3GPP TS 38.300 2022].

No escopo desta dissertação, utilizamos o NS para mapear perfis de tráfego (URLLC/eMBB/mMTC) em BWPs com numerologias específicas, construindo cenários comparáveis que permitem avaliar, de forma controlada, o impacto das métricas de rede sobre o SL (vide seção 3.5.2). Esse enquadramento ajuda a distinguir o efeito

intrínseco da rede daquele induzido pela própria carga do SL, fornecendo evidências mais claras sobre *quais* métricas são mais críticas e *quanto* elas importam.

2.2 Trabalhos Relacionados

Esta dissertação se coloca no cruzamento entre ML — em particular o SL — e redes móveis (5G/B5G). É justamente nesse ponto de interseção que a literatura ainda mostra um vazio: embora existam avanços significativos em cada domínio, os estudos permanecem majoritariamente restritos a um ou outro campo. No SL, prevalece o foco em arquiteturas, compressão e estratégias de corte; Por outro vertente, em 5G NR, destacam-se as propostas de redução de atraso e aumento de confiabilidade. Poucos trabalhos, no entanto, avaliam de forma integrada como métricas de rede impactam diretamente a estabilidade e a eficiência do treinamento distribuído.

A literatura pode ser organizada em três vertentes. Cada uma delas evidencia como técnicas de SL e mecanismos de redes móveis se desenvolvem de forma isolada, mas ainda carecem de integração plena. As Tabelas 2.4, 2.5 e 2.7 apresentam sínteses comparativas de cada linha de pesquisa. pouco explorado entre SL e redes móveis de próxima geração. A literatura evidencia que, embora existam contribuições relevantes em cada vertente, persiste uma lacuna no entendimento de como as métricas de rede impactam diretamente a estabilidade e a eficiência do treinamento distribuído. Ao destacar essa ausência, a presente dissertação oferece uma análise integrada, conectando métricas de comunicação a resultados concretos de aprendizado, em um cenário realista de B5G/6G.

Cada vertente evidencia como técnicas de SL e mecanismos de redes móveis se desenvolvem de forma isolada, mas ainda carecem de integração plena. As Tabelas 2.4, 2.5 e 2.7 apresentam sínteses comparativas de cada linha de pesquisa.

2.2.1 Linha ML: Redução de Custo de Treinamento e Tráfego no SL

Trabalhos pioneiros como [Vepakomma et al. 2018] introduziram o SL com o objetivo de reduzir a carga computacional local e proteger dados sensíveis. Desde então, diferentes autores propuseram variações que exploram corte em camadas distintas, atualizações parciais de pesos ou compressão de gradientes. A contribuição desses estudos é significativa para tornar o SL mais leve em cenários de dispositivos limitados, mas a maioria deles assume um ambiente de conectividade estável, sem variabilidade de rede ou falhas de transmissão. Essa premissa simplificada limita a utilidade prática quando o sistema é implantado em redes móveis reais, onde *jitter*, PLR e variação de vazão podem ser incontroláveis, em certas medidas.

Pesquisas posteriores avançaram sobre aspectos de eficiência. O trabalho de [Xu et al. 2024] demonstram como acelerar o SL federado sobre redes sem fio, combinando seleção de *cutlayer* e atualização local para reduzir atrasos de sincronização. O artigo de [Liu, Deng e Mahmoodi 2023] propuseram um modelo híbrido que une SL e aprendizado federado, explorando complementaridades entre custo computacional e sobrecarga de tráfego. Além disso, [Lin et al. 2024] destacam a relevância do SL em cenários de borda no contexto 6G, incluindo variação de numerologias e múltiplos BWP, embora sem avaliar impacto direto de perdas e retransmissões. De forma paralela, [Duan et al. 2022] integram as duas abordagens em *edge computing*, reforçando a importância de sistemas adaptativos. Em todos os casos, nota-se um denominador comum: os resultados surgem de simulações controladas, sem a influência de uma pilha de rede realista.

A Tabela 2.4 resume algumas dessas propostas iniciais. Embora úteis, elas analisam apenas técnicas isoladas e sob condições de rede ideais. Essa lacuna motiva a análise do segundo eixo da literatura, que aborda mecanismos do 5G NR voltados à redução de latência e variabilidade.

Tabela 2.4: Trabalhos em SL focados em redução de custo (sem rede realista)

Autor/Ano	**Técnica**	**Limitações**
[Vepakomma et al. 2018]	SL básico	Não considera métricas de rede
[Ayad, Renner e Schmeink 2021]	SFL, EPSL	Assume conectividade perfeita
[Duan et al. 2022]	Integração SL+FL	Cenários ideais de conectividade
[Wu et al. 2023]	*cutlayer* dinâmico	Sem análise de *jitter*/PLR
[Liu, Deng e Mahmoodi 2023]	Modelo híbrido SL+FL	Pouca avaliação sob tráfego real
[Xu et al. 2024]	SL federado sobre redes	Resultados sob simulações simplificadas
[Lin et al. 2024]	SL em 6G de borda	Não inclui perdas/retransmissões

2.2.2 Linha NR: Mecanismos de Redução de Latência e Variabilidade

A evolução do 5G/5G NR introduziu mecanismos como *Configured Grant* (CG), *Semi-Persistent Scheduling* (SPS) e numerologias flexíveis para reduzir atraso de transmissão e melhorar confiabilidade. Esses avanços técnicos respondem diretamente

à necessidade de tráfego com requisitos rígidos de latência, como *Extended Reality* (XR), URLLC e aplicações industriais. Contudo, a literatura ainda raramente conecta tais mecanismos às cargas específicas de aprendizado distribuído, o que cria uma oportunidade entre avanços de rede e demandas de ML.

Estudos recentes evidenciam essa lacuna. No estudo de [Lagén et al. 2023] analisam QoS em tráfego XR no 5G NR, destacando estratégias de priorização que reduzem perdas perceptíveis. Enquanto, [Larrañaga et al. 2023] exploraram CG em cenários URLLC, com ênfase na simulação realista via 5G-LENA, mas sem integração a fluxos de aprendizado. O [Koutlia et al. 2023] investiga escalonadores para tráfego sensível, mostrando ganhos consistentes em XR. [Liu et al. 2023] e [Ali et al. 2021] avançaram em *sidelink* e *Vehicle-to-Everything* (V2X), reforçando a relevância de latência reduzida em redes veiculares. Apesar disso, todos os trabalhos permanecem centrados no desempenho da rede, sem avaliar impacto em *pipelines* de SL.

A Tabela 2.5 sistematiza esses mecanismos e suas limitações na literatura, e a Tabela 2.6 detalha quais abordagens ainda carecem de integração com cenários de aprendizado.

Tabela 2.5: Mecanismos do 5G/NR e lacunas em integração com ML

Mecanismo	Efeito	Limitação na literatura
Configured Grant	Reduz RTT e sinalização	Pouca análise em cenários ML
Numerologias flexíveis	Ajustam TTI e cobertura	*Trade-offs* pouco avaliados com SL
QoS/*scheduling* (XR, URLLC)	Priorização de tráfego sensível	Foco em redes, não em aprendizado

Tabela 2.6: Mecanismos do NR que reduzem latência, mas cuja integração com ML ainda é pouco explorada

Técnicas (NR)	[Lagén et al. 2023]	[Larrañaga et al. 2023]	[Koutlia et al. 2023]	[Ali et al. 2021, Liu et al. 2023]	Esta dissertação
(e) *Configured Grant*	✓	✓			✓
(f) Numerologias flexíveis	✓	✓			✓
(g) QoS/escalonamento para tráfego sensível à latência (XR/URLLC)	✓		✓	✓	✓

2.2.3 Ponte ML–NR: Onde os Domínios se Encontram

Poucos estudos analisam explicitamente a interação entre escolhas de *cutlayer* e mecanismos do 5G NR. Essa interdependência é crítica: cortes precoces aumentam o tráfego de ativações, mas mecanismos como CG ou numerologias mais elevadas podem compensar parte do custo. Cortes mais profundos reduzem o volume de dados transmitidos, mas podem ser inviáveis em dispositivos IoT limitados.

Propostas recentes indicam caminhos parciais. A metodologia de [Thapa et al. 2022] introduz o conceito de *splitfed*, em que decisões de corte e agregação são feitas em conjunto, mas sem avaliação sob variabilidade de rede. O trabalho de [Shiranthika et al. 2023] avança ao estudar resiliência a perdas de pacotes, mostrando que a seleção de corte pode mitigar ou agravar impactos de PLR, embora ainda sem mapear efeitos sobre mecanismos de 5G NR. O estudo realizado por [Wang et al. 2023] investiga o equilíbrio entre privacidade e eficiência em SL de larga escala, tema diretamente associado ao uso de QoS e *slicing* para mitigar sobrecargas.

A Tabela 2.7 sintetiza essas interações de forma geral. No entanto, ainda se observa que a literatura carece de um estudo integrado capaz de avaliar simultaneamente SL, variabilidade de rede e mecanismos avançados de 5G NR. É nesse ponto que esta dissertação contribui, demonstrando em cenários B5G/6G como métricas de latência, vazão e energia afetam a estabilidade do treinamento distribuído.

Tabela 2.7: Interação entre técnicas de SL e mecanismos NR

Técnica (SL)	**Mecanismo (NR)**	**Efeito sobre SL**
Compressão/*Dropout*	QoS + *slicing*	Menor tráfego, mas risco de perda de acurácia
cutlayer precoce	*Configured Grant*, Numerologia alta	Reduz RTT/iteração e tempo de espera para concessão; mitiga aumento de ativações
Mini-batching /atualização local	QoS para tráfego sensível	Estabiliza acurácia e reduz frequência de troca

Como resumo integrado, a revisão evidencia que o estado da arte ainda trata de forma fragmentada a relação entre aprendizado distribuído e redes móveis. Trabalhos em SL concentram esforços em compressão, atualização local e escolha da *cutlayer*, mas assumem redes ideais e ignoram métricas críticas como latência variável, PLR ou energia consumida. Pesquisas em 5G NR, por sua vez, avançam em mecanismos de redução de atraso e aumento de confiabilidade, como CG, numerologias flexíveis e políticas de QoS, mas analisam apenas tráfego sintético ou aplicações multimídia, sem cargas reais de ML.

A comparação das duas vertentes mostra um desalinhamento metodológico: enquanto a comunidade de ML adota cenários laboratoriais simplificados, a de redes prioriza otimizações de camada física e *Media Access Control* (MAC) sem considerar a sobrecarga de gradientes e ativações. Poucos trabalhos exploram de forma integrada os impactos de *cutlayer*, atualização local e políticas de escalonamento em condições reais de rede. Essa ausência fragiliza a aplicabilidade prática das soluções propostas, pois desconsidera justamente o ambiente em que SL será implantado: redes móveis heterogêneas, densas e sujeitas a falhas.

Iniciativas recentes como o projeto *openranbr* e a plataforma *ns-3 Open RAN integration* (ns-O-RAN) ampliam a visão de abertura e programabilidade de redes, oferecendo ferramentas para experimentação de arquiteturas avançadas. Entretanto, ainda não conectam explicitamente métricas de rede com estabilidade de treinamento em SL, deixando uma lacuna importante para avaliação científica.

Portanto, a lacuna persiste: é necessário avaliar de forma integrada como latência, *jitter*, PLR, vazão e consumo energético afetam o desempenho do SL em cenários realistas B5G/6G. É exatamente essa a contribuição central desta dissertação, que conecta avanços de 5G NR a modelos de SL, preenchendo o espaço entre teoria e prática.

CAPÍTULO 3

Proposta de Arquitetura de Split Learning Orientada a Métricas de Rede para Desempenho de Redes Móveis B5G/6G

Este capítulo descreve o cenário experimental, a estrutura do modelo de SL e a modelagem da rede 5G no simulador ns-3. O objetivo é demonstrar como os blocos foram integrados em um fluxo reprodutível e quais decisões metodológicas orientaram a configuração do sistema, conectando escolhas de projeto às métricas analisadas nos resultados (latência, *jitter*, vazão, PLR e energia).

3.1 Cenário do Sistema Proposto

Com o intuito de atingir os objetivos dessa proposta, foi realizada a implementação do SL, denominada *SplitLearning-ns3* [LABORA-INF/UFG 2025], incorporando as funcionalidades requeridas. O cenário experimental adota 102 *User Equipments* (UEs) conectados a duas gNBs, refletindo uma topologia de célula densa característica de redes B5G/6G. O número de UEs foi definido com base em estudos de escalabilidade em ns-3/5G-LENA: valores inferiores não evidenciam de forma adequada a competição entre fluxos, enquanto valores muito superiores tornam o tempo de simulação proibitivo e comprometem a coleta sistemática de métricas. Nesse contexto, a escolha de 102 UEs estabelece um ponto de equilíbrio entre realismo experimental e viabilidade computacional.

A Figura 3.1 não apenas ilustra a arquitetura proposta, mas também evidencia a relação de causa–efeito entre seus elementos. Os clientes (1) geram tráfego e executam as camadas neurais locais (2), o que reduz a carga computacional do servidor, mas aumenta a dependência da rede para envio de ativações. O servidor de aprendizado (3) recebe essas ativações e processa os gradientes, de modo que atrasos ou perdas no canal comprometem a sincronização com os clientes. A interface ns3-ai (4) atua como elo crítico entre simulação e aprendizado, garantindo o fluxo eficiente de dados; falhas nessa troca amplificam o impacto da rede sobre a convergência do modelo. O módulo de simulação ns-3 (5) injeta

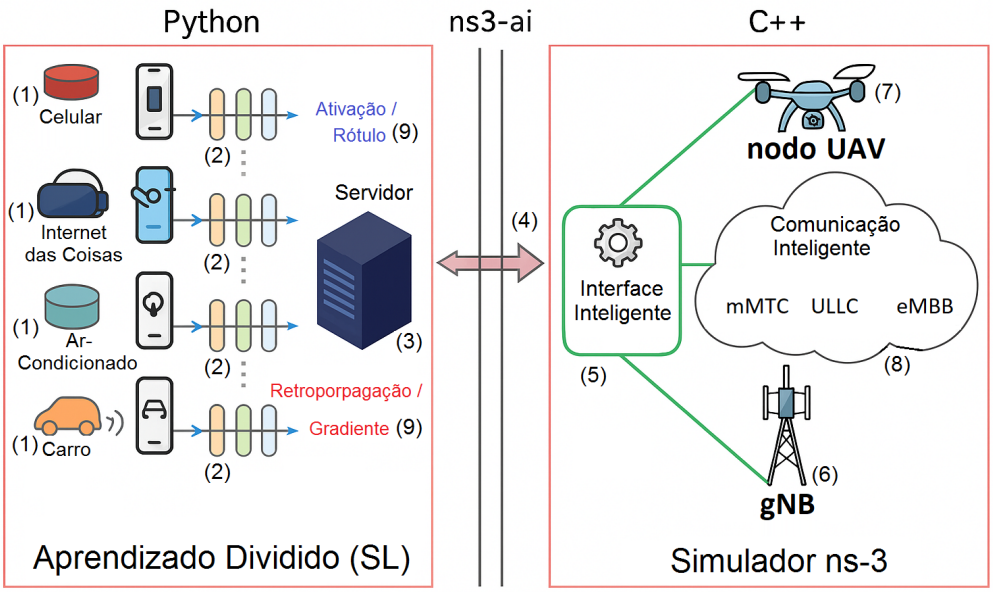

Figura 3.1: Arquitetura SL integrando ns-3/5G-LENA com ns3-ai. Elementos numerados: (1) Dispositivos cliente; (2) Camadas neurais do cliente; (3) Servidor de aprendizado; (4) Interface ns3-ai; (5) Módulo de simulação ns-3; (6) gNB; (7) Nó UAV; (8) Nuvem de comunicação inteligente (mMTC, URLLC, eMBB); (9) Ciclo de treinamento de ativações e gradientes.

variabilidade realista (latência, *jitter*, perdas, energia), permitindo que o gNB (6) seja avaliado como ponto central de alocação de recursos, em que congestionamentos ou políticas de escalonamento afetam múltiplos clientes. O nó *Unmanned Aerial Vehicle* (UAV) (7) adiciona mobilidade e variação espacial, ampliando a heterogeneidade do cenário. A nuvem de comunicação inteligente (8), composta por *slices* mMTC, URLLC e eMBB, estabelece diferentes prioridades de QoS, de modo que a priorização de um perfil pode degradar o desempenho de outro, refletindo os *trade-offs* típicos dos cenários que consideram *network slicing*. Por fim, o ciclo de treinamento de ativações e gradientes (9) fecha o laço entre rede e aprendizado: atrasos, perdas ou variabilidade de tráfego repercutem diretamente na acurácia e no tempo de convergência. Essa integração reforça o propósito central da dissertação, ao mostrar como métricas de rede se traduzem em impacto direto sobre a robustez do SL.

Nota metodológica. O nó UAV (elemento 7) é apenas ilustrativo de aplicações típicas do 5G e não foi utilizado nos experimentos.

Posicionamento e distância UE-gNB

A área de simulação possui dimensões da ordem de centenas de metros por lado, com UEs fixos em anéis concêntricos a diferentes distâncias do gNB mais próximo. A distância euclidiana entre um UE em $P_1(x_1, y_1, z_1)$ e o gNB em $P_2(x_2, y_2, z_2)$ é calculada

Tabela 3.1: Parâmetros de simulação utilizados neste estudo

Parâmetro	**Valor**
Número de gNB	2
Número de UE	102
Largura de banda (*carrier*)	100 MHz
Numerologia BWP1	$\mu = 4$
Numerologia BWP2	$\mu = 2$
Tempo de simulação	10 s
Dataset de treino	*MNIST* (CNN)
Métrica de energia (*proxy*)	$E = P \cdot d^2 \cdot t$

como:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}. \tag{3-1}$$

Esse controle espacial permite avaliar a influência da distância em atraso, perda de pacotes e consumo energético, mitigando efeitos de borda e interferência intercelular.

Parametrização e escala do cenário

Os principais parâmetros de execução são expostos via linha de comando, destacando: `--gNbNum`, `--ueNumPerGnb` e pesos por *slice*. O total de usuários é dado por

$$N_{\text{ue}} = \texttt{gNbNum} \times \texttt{ueNumPerGnb}.$$

Exemplo de execução:

```
./ns3 run scratch/SplitLearning-B5G/cttc-nr-split
   -gNbNum=2 -ueNumPerGnb=51 -UrllcWeight=5
   -EmbbWeight=1 -MmtcWeight=1
```

Além da variação da distância UE–gNB, foram testados diferentes perfis de atenuação (*pathloss offset*) para calibrar a severidade do canal: *Low* (2 dB), *Moderate* (3 dB) e *High* (4 dB). Esses perfis modulam atraso, *jitter* e PLR sem alterar a topologia, permitindo análise controlada.

Resumo dos parâmetros experimentais. Para tornar explícitas as configurações-base utilizadas ao longo dos cenários, a Tabela 3.1 consolida os principais parâmetros de simulação.

Artefatos e reprodutibilidade

As simulações foram conduzidas no arquivo `cttc-nr-split.cc`, responsável pela configuração do cenário no ns-3. Resultados e métricas são exportados em formato `CSV` no diretório `plots/`, garantindo repetibilidade e comparação justa entre cenários.

Como panorama geral, a configuração do cenário e os elementos representados na Figura 3.1 reforçam a integração entre variabilidade de rede e comportamento do SL. Essa perspectiva será retomada no Capítulo 4, ao demonstrar como métricas específicas — latência, vazão, *jitter*, PLR e consumo energético — moldam a estabilidade e a eficiência do treinamento distribuído.

3.2 Estrutura do Modelo SL

O modelo de aprendizado adotado foi uma CNN particionada com corte intermediário: as camadas iniciais residem no cliente e as finais no servidor. A notação e as dependências de comunicação por iteração deixam claro por que o desempenho do SL é sensível às condições de rede.

Notação e equações do fluxo SL. Para um cliente i com (x_i, y_i):

$$z_i = f_c(x_i; w_c^i), \qquad \hat{y}_i = f_s(z_i; w_s), \tag{3-2}$$

$$\mathcal{L}_i = \mathcal{L}(\hat{y}_i, y_i), \qquad g_i = \frac{\partial \mathcal{L}_i}{\partial z_i}. \tag{3-3}$$

A ativação intermediária z_i é enviada ao servidor; o gradiente g_i retorna ao cliente para atualização local $w_c^i \leftarrow w_c^i - \eta \nabla_{w_c^i} \mathcal{L}_i$. Portanto, cada iteração depende de (i) ida de z_i e (ii) volta de g_i, diretamente impactadas por latência, *jitter*, vazão e PLR.

Arquitetura em camadas particionadas. A CNN adotada neste trabalho foi estruturada em três blocos principais: `ml_model_in` (cliente), `ml_model_hidden` e `ml_model_out` (servidor). No lado do cliente, concentram-se as camadas convolucionais iniciais e operações de ativação, responsáveis por extrair padrões elementares dos dados. Esse desenho reduz a carga computacional local e preserva a privacidade, mas tem como consequência direta o aumento da dependência da rede para transmissão das ativações. No lado do servidor, por sua vez, localizam-se as camadas convolucionais mais profundas e as camadas densas, que consolidam a classificação. Esse arranjo amplia a capacidade de generalização do modelo, mas implica maior sensibilidade a perdas e atrasos de comunicação. Portanto, a escolha do ponto de corte (*cut layer*) gera um equilíbrio entre custo computacional no cliente e volume de ativações trafegadas na rede, sendo decisiva para a robustez do SL em cenários móveis heterogêneos.

A Figura 3.2 sintetiza graficamente essa organização, destacando como a divisão cliente–servidor da CNN impacta o fluxo de ativações e gradientes. A visualização reforça a lógica discutida: cortes mais precoces aliviam o dispositivo, mas aumentam o tráfego de rede; cortes mais profundos reduzem comunicações, mas sobrecarregam o cliente. Assim,

a figura não apenas ilustra a arquitetura proposta, como também contextualiza a relevância metodológica de investigar o SL sob condições realistas de rede.

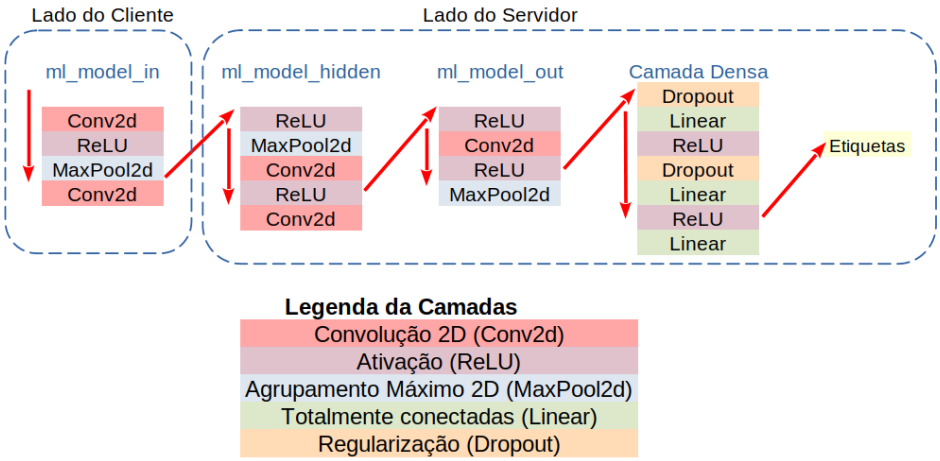

Figura 3.2: CNN particionada: a divisão cliente–servidor define o balanço entre custo local e tráfego de ativações, modulando a resiliência do treinamento em condições variáveis de rede.

Justificativa do ponto de corte. Cortes precoces reduzem computação local, mas inflacionam $\|z_i\|$ (maior demanda de vazão, maior sensibilidade a PLR/*jitter*); cortes tardios exigem clientes mais potentes. Adotamos um corte intermediário (após o segundo bloco convolucional), equilibrando custo no cliente e volume de dados trafegados, e isolando o efeito da rede na convergência.

Estratégias de eficiência de comunicação. Quatro estratégias foram usadas sem alterar a essência do SL: (1) *Dropout* nas camadas superiores (melhora generalização e reduz densidade de ativações), (2) *local update steps* (diminuem a frequência de trocas cliente–servidor), (3) *batching* controlado (amortiza latências fixas por mensagem) e (4) *computation offloading* (delegação de camadas intermediárias ao servidor).

Pipeline de ML, dataset e comunicação. O treinamento usa MNIST como conjunto de dados local por cliente (10 classes), carregado via `torchvision.datasets.MNIST` e `DataLoader` com *batching*. A função de perda é `nn.CrossEntropyLoss`; o otimizador no cliente é *Stochastic Gradient Descent* (SGD). A comunicação cliente–servidor é feita por *sockets* (endereço 127.0.0.1, porta 19089), com envio das ativações z e retorno do gradiente g a cada mini-lote. Monitoramos `total_comm_time` e `total_comm_data` para quantificar sobrecarga. Bibliotecas utilizadas: `numpy`, `pandas`, `torch`/`torchvision`, `matplotlib`, `tqdm`.

Algoritmo 3.1 Fluxo de Treinamento em SL – sequência cliente/servidor e retropropagação

Input: Modelo CNN M, ponto de corte L_c, dados de treino D

for *cada época* e **do**
  for *cada mini-lote* $b \in D$ **do**
    Cliente executa camadas $M[1{:}L_c]$ sobre b e produz z
    Envia z ao servidor (via ns3-ai/ns-3)
    Servidor executa $M[L_c{+}1{:}\text{fim}]$ e produz $\hat{y}$
    Servidor calcula $\mathcal{L}(\hat{y}, y)$, retropropaga e envia $g = \partial\mathcal{L}/\partial z$
    Cliente atualiza pesos de $M[1{:}L_c]$ com g

Detalhamento. O Algoritmo 3.1 materializa o SL ao dividir a rede de simulação no ponto L_c: no cliente, as camadas iniciais extraem representações do mini-lote b e produzem as ativações z (linha 3); em seguida, apenas z (e não os dados brutos) é enviado ao servidor (linha 3), o que sinaliza um ganho de privacidade por reduzir a exposição do dado original [Vepakomma et al. 2018]. No servidor, o restante do modelo gera a predição $\hat{y}$ (linha 3) e calcula-se a perda com retropropagação, devolvendo o gradiente $g = \partial\mathcal{L}/\partial z$ ao cliente (linha 3), que então atualiza apenas os pesos "antes do corte" (linha 3). Esse encadeamento cria um *trade-off* prático: escolher um L_c raso tende a produzir ativações maiores (mais *bytes* em trânsito), o que alivia o cômputo local mas amplia a sobrecarga de rede; por outro lado, um L_c profundo comprime melhor a informação (menos dados a transmitir) à custa de mais processamento no dispositivo. Em ambientes reais, a latência, a banda disponível e o *jitter* da conexão impactam diretamente o tempo por época e a convergência, pois cada mini-lote adiciona um ciclo "envia z / recebe g" [Kurose 2021]. Ao combinar perdas canônicas (`CrossEntropyLoss`) e otimizadores padrão (SGD) no *PyTorch*, sobre um conjunto clássico como o MNIST, o *pipeline* isola de forma limpa o efeito da rede: métricas como `total_comm_time` e `total_comm_data` refletem a parcela "rede" do custo total, permitindo comparar configurações de L_c, tamanhos de lote e condições de enlace. Em resumo, o SL exige co-projeto entre aprendizado e comunicação: decisões de ML (onde cortar, como otimizar) e "de rede" (atraso, vazão, variabilidade) se entrelaçam e, se mal ajustadas, tornam o treinamento lento ou instável [Vepakomma et al. 2018, Kurose 2021].

3.3 Fluxo de Trabalho em Seis Etapas

O ciclo de treinamento do SL integrado ao ns-3 segue seis etapas:

1. Configuração da rede 5G-LENA (gNBs, UEs), *slices* (URLLC, eMBB, mMTC), largura de banda e numerologias (por BWP).
2. Inicialização do modelo de aprendizado e definição da *cut layer*.

3. Envio das ativações z dos clientes ao servidor.
4. Propagação direta e retropropagação no servidor sobre a parte superior do modelo.
5. Retorno dos gradientes g do servidor aos clientes.
6. Atualização local dos parâmetros no cliente, completando a etapa de treinamento.

Esse fluxo garante sincronização entre múltiplos dispositivos e o servidor, respeitando as restrições de comunicação impostas pela simulação.

3.4 Reprodutibilidade e Instrumentação

Para assegurar reprodutibilidade, os experimentos são parametrizados (topologia, BWP/numerologia por *slice*, arquitetura do modelo, *batch size*, épocas). A instrumentação exporta arquivos em formato `CSV` contendo os seguintes registros:

- Rede: latência, vazão, *jitter*, PLR, consumo energético, distância, identificação de *slice*/BWP;
- Aprendizado: acurácia de validação, tempo de treinamento; métricas de comunicação como `total_comm_time` e `total_comm_data`.

Esses registros permitem análises entre condições de rede e desempenho do SL, garantindo transparência metodológica e comparabilidade entre cenários.

Para complementar essa descrição textual e oferecer uma visão resumida, a Tabela 3.2 reúne os principais parâmetros de saída da simulação. Enquanto as seções seguintes detalham formalmente cada métrica (com definições matemáticas e análises específicas), a tabela funciona como referência consolidada dos indicadores observados, reforçando a clareza e fechando a parte metodológica da instrumentação.

Tabela 3.2: Parâmetros de saída da simulação

Parâmetros	**Descrição**	**Observação**
Energia (J)	Energia consumida durante a comunicação/processamento	Joules
Latência (s)	Tempo para um dado percorrer de um ponto a outro	Segundos
PLR (%)	Número de pacotes perdidos em relação ao total transmitido	Percentual
Posições (m)	Posicionamento dos dispositivos em relação ao gNB, incluindo distâncias	Metros
Vazão (Mbps)	Quantidade de dados transferidos — vazão efetiva da rede	Megabits por segundo

A presença consolidada desses parâmetros na Tabela 3.2 é mais do que um simples registro técnico: ela estabelece o elo metodológico entre o que foi configurado no simulador e os indicadores que sustentam as análises dos capítulos seguintes. Cada métrica cumpre um papel específico na avaliação integrada entre rede e aprendizado. A vazão, por exemplo, reflete diretamente a eficiência do canal de comunicação, enquanto a latência e o *jitter* indicam estabilidade temporal essencial para preservar o ritmo de convergência do modelo de SL. Da mesma forma, a taxa de perda de pacotes atua como um termômetro da confiabilidade do enlace, impactando de forma imediata a acurácia de validação. Porém o consumo energético, ainda que não altere diretamente a curva de aprendizado, representa um fator crítico de viabilidade prática em cenários B5G/6G, sobretudo em dispositivos com restrições de bateria. Por último, a posição dos dispositivos em relação ao gNB assegura que variações espaciais sejam rastreadas, permitindo análises de correlação entre distância, intensidade de sinal e métricas de desempenho.

Assim, a tabela ancora a discussão metodológica em um conjunto de indicadores mensuráveis, conferindo robustez às comparações entre cenários e fortalecendo a reprodutibilidade científica deste trabalho.

A configuração detalhada de hardware e software utilizada nas simulações encontra-se no Apêndice A (ver Tabela A.1).

Reprodutibilidade, Instrumentação e Métricas de Rede

Para garantir a reprodutibilidade, os experimentos foram parametrizados (topologia, BWP/numerologia por *slice*, arquitetura do modelo, *batch size*, épocas), e a instrumentação exporta arquivos `.csv` contendo os principais indicadores. Cada uma é formalizada a seguir, com definições matemáticas que orientam a análise apresentada no Capítulo 4.

As métricas adotadas neste estudo seguem as definições formais do 3GPP para redes 5G — em especial as [3GPP TS 23.501 2025] e [3GPP TS 38.300 2022], que estruturam o modelo de QoS (latência, *jitter*, vazão e confiabilidade/PLR) e a sua materialização na interface 5G NR. Esse alinhamento normativo assegura comparabilidade com a literatura e com estudos de referência. No caso da energia, adotou-se uma formulação amplamente utilizada em avaliações de eficiência de redes móveis [3GPP TS 38.912 2018], em que o consumo total é estimado expresso na Equação (3-8), podendo ser ajustado por fatores de distância/atenuação. Esse modelo reflete a abordagem consolidada em estudos de dimensionamento energético de redes celulares, garantindo que a análise mantenha aderência às práticas aceitas internacionalmente.

Métrica de Latência. latência foi avaliada considerando tanto o valor médio quanto sua variabilidade, obtida a partir da sequência de atrasos x_i observados nos n pacotes transmitidos. A média aritmética da latência é expressa por:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{3-4}$$

enquanto o desvio-padrão amostral, que indica a dispersão em torno da média, é definido como:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \tag{3-5}$$

Essas métricas permitem caracterizar não apenas o tempo médio de resposta da rede, mas também a estabilidade temporal da comunicação, aspecto essencial para avaliar o impacto da infraestrutura sobre o processo de treinamento em SL.

Throughput (Vazão)

O *throughput* representa a taxa efetiva de dados enviados/recebidos pelos dispositivos durante a simulação. Sua definição matemática é dada pela razão entre o total de bits recebidos B_{rx} e a duração da transmissão Δt:

$$Thr = \frac{B_{rx}}{\Delta t}. \tag{3-6}$$

Essa métrica reflete a eficiência da utilização dos recursos da rede. Valores mais altos de vazão indicam maior capacidade de suporte ao treinamento colaborativo, enquanto valores reduzidos podem comprometer a continuidade do processo de aprendizado distribuído, sobretudo em cenários com tráfego heterogêneo.

A direção da medição no contexto do SL é a seguinte: *uplink* (UE→gNB) para o envio de ativações ao servidor e *downlink* (gNB→UE) para o retorno de gradientes na retropropagação. Quando apresentarmos um valor agregado, indicaremos como *throughput* total (UL+DL).

Packet Loss Ratio (PLR)

A PLR é uma métrica essencial para avaliar a confiabilidade da rede, especialmente em cenários de SL, onde a falha na entrega de ativações intermediárias pode comprometer a continuidade do treinamento. A PLR é definida como:

$$PLR = \frac{N_{lost}}{N_{tx}}, \tag{3-7}$$

em que N_{lost} corresponde ao número de pacotes perdidos e N_{tx} ao total de pacotes transmitidos. Diferentemente da latência e do *throughput*, o impacto da PLR é mais severo, pois implica em falhas irrecuperáveis na comunicação cliente–servidor, resultando em degradação direta da acurácia do modelo.

Consumo de Energia

O consumo energético é uma métrica relevante para avaliar a eficiência da rede e dos dispositivos em cenários de SL, especialmente considerando aplicações em ambientes B5G/6G com dispositivos IoT e UEs com recursos limitados. A energia total consumida E_{tot} é dada por:

$$E_{tot} = P_{tx} \cdot t_{on}, \tag{3-8}$$

em que P_{tx} representa a potência de transmissão e t_{on} o tempo de atividade do dispositivo. Essa formulação permite relacionar diretamente a configuração da potência com os custos energéticos associados à comunicação durante o treinamento. Cenários de maior potência tendem a reduzir a latência e melhorar a acurácia, mas também implicam em maior consumo energético.

Nos resultados a seguir, essa métrica será analisada em conjunto com os tipos de tráfego e *slices*, de forma a verificar o *trade-off* entre desempenho da rede, eficiência do treinamento e custo energético associado.

3.4.1 Considerações e Análise Estatística

Esta seção explicita o tratamento estatístico adotado para as métricas de rede e de aprendizado definidas matematicamente na Seção 3.4 (Equações (3-1)–(3-5)). O objetivo é qualificar a interpretação dos resultados do Capítulo 4, adicionando noções de variabilidade, incerteza e comparação entre cenários.

Replicações e aleatorização. Os experimentos foram conduzidos com múltiplas réplicas independentes, empregando sementes distintas para assegurar aleatoriedade controlada e capturar variações inerentes tanto ao simulador quanto ao tráfego *On–Off*. Para cada métrica m (latência, *jitter*, vazão, PLR e energia), reportamos a média $\bar{m}$ e o desvio-padrão s_m ao longo das réplicas, além da média temporal intra-execução quando aplicável.

Intervalos de confiança. Quando indicado nas figuras do Capítulo 4, faixas sombreadas correspondem a *Confidence Intervals* (*CIs*) de 95% para a média (aproximação normal com correção de *t Student* quando n é pequeno). Em séries temporais, *CIs* são mostrados por janela deslizante para evidenciar estabilidade da curva de aprendizado.

Testes de hipótese entre cenários. Para comparar cenários (p. ex., diferentes BWPs, numerologias ou *slices*), utilizamos: (i) teste t de *Welch* (variâncias possivelmente distintas) para duas condições; e (ii) *Analysis of Variance* (ANOVA) de uma via para três ou mais condições, seguida de *post hoc* de *Games–Howell* quando apropriado. Em presença de violações claras de normalidade/heterocedasticidade, adotamos alternativas não paramétricas (*Mann–Whitney* ou *Kruskal–Wallis*).

Tamanho de efeito e múltiplas comparações. Além do p-valor, reportamos o tamanho de efeito (*Cohen's d* para duas amostras; η^2 parcial para ANOVA), que qualifica a relevância prática da diferença. Correções para múltiplas comparações seguem o procedimento de *Benjamini–Hochberg* (controle de *False Discovery Rate* (FDR)).

Boas práticas de reporte. As tabelas e figuras apresentam, quando cabível, $\bar{m} \pm s_m$ e o número de réplicas (n). Para curvas de acurácia/convergência, destacamos medianas e quartis em cenários com assimetria acentuada, reduzindo a influência de *outliers*. Em todos os casos, mantemos a rastreabilidade dos arquivos `.csv` exportados (Seção 3.4).

Limitações. O objetivo principal deste trabalho é comparar tendências entre cenários sob controle experimental. Embora os testes descritos forneçam suporte inferencial, os resultados ainda dependem das suposições do modelo de tráfego e do nível de abstração do *control plane*. Assim, recomenda-se interpretar efeitos marginais com cautela e priorizar tamanhos de efeito na discussão.

Reprodutibilidade. *Scripts* auxiliares (não mostrados) automatizam: (i) aglutinação dos `.csv`, (ii) checagem de normalidade (*Shapiro–Wilk*) e homogeneidade (*Levene*), (iii) cálculo de *CIs* e tamanhos de efeito, e (iv) geração de gráficos com bandas de incerteza. O *seed* e a configuração do cenário são registrados no cabeçalho dos artefatos exportados.

3.5 Estrutura do Modelo 5G

3.5.1 Configuração de Canais

A configuração 5G NR no ns-3/5G-LENA segue uma sequência típica e reproduz as diretrizes do 3GPP para cenários urbanos densos. A etapa de configuração garante não apenas a correta associação entre UEs e gNBs, mas também a parametrização de elementos físicos que influenciam diretamente as métricas avaliadas no SL. Nesse contexto, torna-se relevante explicitar não apenas os valores configurados (já sumarizados na Tabela 3.1), mas também os efeitos qualitativos que cada decisão acarreta sobre o desempenho do SL em ambientes B5G/6G (ver Tabela 3.3).

A Tabela 3.3 sintetiza tais impactos, fornecendo uma perspectiva interpretativa adicional que conecta parâmetros físicos da rede com potenciais efeitos no ciclo de treinamento distribuído.

Os parâmetros listados na Tabela 3.3 não atuam de forma isolada, mas se combinam na cadeia de configuração. Por exemplo, a escolha da potência de transmissão e da numerologia impacta diretamente a confiabilidade do enlace e a eficiência energética; já a definição do perfil de tráfego por *slice* determina a prioridade e a granularidade com que os pacotes alimentam o ciclo de treinamento do SL. Esses fatores, quando aplicados em conjunto ao modelo de propagação urbano denso, explicam por que certos cenários

Tabela 3.3: Aspectos qualitativos de parâmetros 5G/B5G e seus impactos no SL

Parâmetro 5G/B5G	Impacto qualitativo no SL
Modelo de Propagação 3GPP	Condições urbanas densas elevam PLR e podem comprometer a acurácia do SL; cenários moderados permitem equilíbrio entre confiabilidade e consumo. A distância UE–gNB é determinante para *trade-offs* entre latência e energia.
Numerologia por BWP	Numerologias elevadas (*slots* curtos) favorecem baixa latência e estabilidade do SL em URLLC; numerologias moderadas maximizam *throughput* em eMBB; numerologias baixas reduzem consumo em mMTC.
Perfil de Tráfego por *Slice*	Pacotes pequenos e frequentes (URLLC) reduzem *jitter* mas elevam *overhead*; pacotes grandes (eMBB) aumentam eficiência espectral; transmissões esparsas (mMTC) preservam energia mas limitam taxa de atualização do modelo.
Potência de Transmissão (*TxPower*)	Potências maiores reduzem perdas de enlace e beneficiam a convergência em cenários URLLC; contudo, ampliam consumo energético e interferência *inter-slice*, prejudicando a eficiência em mMTC.

apresentam maior latência ou maior PLR, refletindo em estabilidade reduzida e menor acurácia final.

1. Criação de nós (`NodeContainer`) para gNBs e UEs; posicionamento fixo para análise controlada por distância;
2. Instalação de dispositivos: `NrHelper::InstallGnbDevice()` e `NrHelper::InstallUeDevice()`;
3. Parâmetros físicos: frequência central e largura de banda por BWP (ex.: `CcBwpCreator::SimpleOperationBandConf`); potência via `ns3::NrGnbPhy::TxPower`;
4. Associação UEs–gNBs com `AttachToClosestEnb()` e ativação de *Evolved Packet Core* (EPC)/pilha *Internet Protocol* (IP) quando aplicável;
5. Canal/propagação: `ns3::ThreeGppChannel` e modelos coerentes com cenário urbano denso.

A enumeração acima detalha a ordem de execução dentro do simulador, mas é na sua articulação sequencial que se observa o vínculo causa–efeito. Cada etapa adiciona restrições ou condições que moldam o espaço de possíveis resultados: a criação dos nós define a topologia; a instalação de dispositivos habilita a comunicação; a configuração física ajusta capacidade e consumo; a associação define quem compete por recursos; e

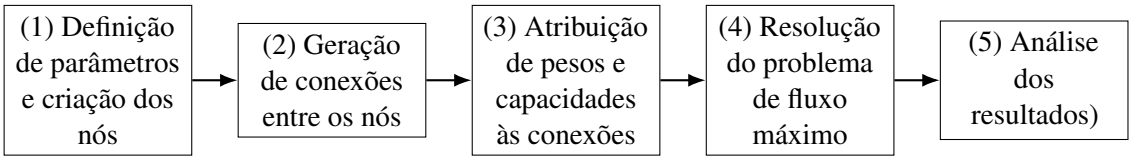

Figura 3.3: Configuração de canais - (1) parâmetros 5G/NR e topologia → (2) conexões e associação UE–gNB → (3) pesos/capacidades (BWP, numerologia, QoS) → (4) alocação/otimização de fluxo → (5) métricas (latência, vazão, *jitter*, PLR, energia) e impacto no SL.

os modelos de propagação fixam as perdas reais de canal. Essa progressão explica por que determinados *slices* (como URLLC) se beneficiam mais de parâmetros agressivos de latência, enquanto outros (como mMTC) necessitam de compromissos energéticos.

A Figura 3.3 integra os componentes anteriores em uma cadeia lógica. Do ponto (1) ao (3), definem-se as condições iniciais do enlace e as prioridades de tráfego; em (4), o escalonador resolve o problema de fluxo sob restrições de capacidade; finalmente, em (5), emergem as métricas de rede que retroalimentam o desempenho do SL. Essa representação demonstra que cada parâmetro configurado afeta não apenas um aspecto isolado, mas todo o processo de treinamento distribuído. Assim, a PLR surge como variável crítica para a acurácia, enquanto a latência e o *jitter* modulam a velocidade de convergência, e a vazão estabelece o limite de *throughput* efetivo de gradientes e ativações. A conjunção desses efeitos justifica as tendências observadas nos resultados experimentais do Capítulo 4, onde se confirmam os vínculos antecipados nesta subseção.

3.5.2 Fatiamento com BWPs e Numerologias

O fatiamento (NS) foi implementado por meio de BWPs com numerologias distintas, refletindo requisitos heterogêneos de QoS em 5G NR. Um BWP corresponde a um conjunto contíguo de *resource blocks* sobre uma portadora e pode ter configuração própria (largura, numerologia, canais de controle/dados), havendo um BWP primário para o anexo inicial e BWPs secundários selecionáveis via RRC ao longo da simulação [Koutlia e al. 2024, Luna, Oliveira e Silva 2021]. Em 5G NR, a numerologia μ define o espaçamento entre subportadoras e a granularidade temporal:

$$\Delta f(\mu) = 15\ \text{kHz} \cdot 2^{\mu} \qquad \text{e} \qquad T_{\text{slot}}(\mu) = \frac{1\ \text{ms}}{2^{\mu}}, \tag{3-9}$$

com tipicamente 14 símbolos *Orthogonal Frequency Division Multiplexing* (OFDM) por *slot* (*Cyclic Prefix* (CP)). Numerologias maiores reduzem T_{slot} (e viabilizam *minislots*), favorecendo baixa latência/alta confiabilidade, ao custo de maior sensibilidade a propagação [Medeiros, Santos e Almeida 2022, Koutlia e al. 2024].

Diretrizes de mapeamento *Slice* → BWP/Numerologia. Para capturar os requisitos clássicos 3GPP e isolar efeitos de rede sobre o SL, adotamos a seguinte diretriz (ajustada por cenário):

- eMBB: BWP com maior largura útil e numerologia moderada ($\mu \in \{1,2\}$; Δf = 30∼60 kHz), priorizando *throughput* agregado e eficiência espectral;
- URLLC: BWP com numerologia mais alta ($\mu \in \{2,3\}$; Δf = 60∼120 kHz), permitindo *mini-slots* curtos, baixa latência e alta confiabilidade de enlace;
- mMTC: BWP enxuto, numerologia baixa ($\mu = 0$; Δf = 15 kHz), tráfego esparso com foco em cobertura/robustez e eficiência energética [Luna, Oliveira e Silva 2021, Koutlia e al. 2024, Medeiros, Santos e Almeida 2022].

Os UEs são agrupados por *slice* e mapeados aos respectivos BWPs; parâmetros de envio (intervalo e tamanho de pacotes) são ajustados ao perfil de serviço [CTTC 2023, Sousa 2022].

3.5.3 Classes de QoS e Identificadores 5QI

No escopo do 3GPP, os requisitos de QoS são formalizados por meio dos identificadores *5G QoS Identifier* (5QI). Cada identificador traduz, de forma padronizada, um conjunto de atributos-chave que orientam o tratamento do tráfego, incluindo a latência fim a fim alvo, a PLR admissível e o nível de prioridade associado. Essa taxonomia atua como elo entre o plano conceitual e a configuração prática, garantindo que diferentes perfis de serviço sejam tratados de forma consistente na rede.

Na prática, os 5QIs se alinham ao mapeamento *Slice*→BWP/numerologia apresentado na Seção 3.5.2. Assim, serviços sensíveis ao atraso, como o URLLC, recebem identificadores de baixa latência e alta confiabilidade; aplicações multimídia e de alto volume de tráfego, típicas do eMBB, associam-se a identificadores com maiores taxas de dados e maior tolerância a atraso; já os cenários massivos de IoT, característicos do mMTC, priorizam eficiência energética e ampla cobertura, ainda que com restrições de taxa de atualização.

Tabela 3.4: Exemplos ilustrativos de 5QI e requisitos típicos (conforme diretrizes 3GPP)

5QI	Serviço típico	Latência alvo (ms)	PLR alvo
1	Voz conversacional	100	10^{-2}
2	Vídeo ao vivo	150	10^{-3}
3	URLLC crítico	5	10^{-5}
9	Dados em massa (eMBB)	300	10^{-6}

A Tabela 3.4 exemplifica como diferentes serviços se materializam em parâmetros normativos. O 5QI 1, associado à voz conversacional, reflete requisitos moderados de

latência e perdas, típicos de chamadas interativas. Já o 5QI 3, voltado para cenários de URLLC, explicita a necessidade de latência ultrabaixa e perdas residuais mínimas, condição essencial para aplicações críticas. O identificador 9, por sua vez, ilustra o perfil eMBB, em que o foco é maximizar a vazão mesmo com latências mais longas, característica de fluxos massivos de dados. Esses exemplos não esgotam a lista definida pelo 3GPP, mas sintetizam a lógica de especialização que sustenta a arquitetura de rede baseada em *network slicing*.

No restante do texto, esse enquadramento é utilizado como referência conceitual. A instrumentação de simulação (Seção 3.4) avalia diretamente latência, *jitter*, vazão, PLR e consumo energético, enquanto os 5QIs fornecem o pano de fundo normativo para interpretar os resultados de cada *slice*/BWP. Dessa forma, garante-se que a análise mantenha aderência às diretrizes internacionais, ao mesmo tempo em que se preserva a fidelidade experimental do modelo proposto.

Modelagem de Tráfego On–Off por Slice

Para capturar intermitência realista, utilizamos geradores *On–Off* específicos por *slice*:

- eMBB: períodos ativos mais longos e taxas médias elevadas (p. ex., pacotes de 500 B em 8–12 pps), representando fluxos de alto consumo de dados;
- URLLC: pacotes pequenos (p. ex., 32–64 B) e intervalos ativos curtos/altas frequências (15–30 pps), refletindo restrição de atraso e confiabilidade;
- mMTC: tráfego esporádico de baixo volume (1–3 pps, pacotes reduzidos), típico de telemetria IoT [Medeiros, Santos e Almeida 2022, Villegas e Costa 2024].

Combinada ao fatiamento por BWP/numerologia, essa modelagem evidencia os compromissos entre latência, confiabilidade e eficiência espectral. A instrumentação por *FlowMonitor* coleta atraso, *jitter*, vazão, PLR e energia por fluxo e por *slice*.

Associação de UEs, containers e instrumentação

A associação por *slice* foi operacionalizada via containers dedicados (p. ex., `UEVoiceContainer` → tráfego de voz/eMBB; `UELowLatContainer` → URLLC), com mapeamento 1–para–1 para BWPs. Essa organização simplifica: (i) a aplicação de perfis de tráfego distintos, (ii) o ajuste de parâmetros (tamanho/intervalo de pacotes) e (iii) a coleta segregada de métricas por *slice*/BWP [CTTC 2023, Sousa 2022]. Quando necessário, um escalonador customizado (`MyCustomScheduler`) pondera prioridades por *slice* no `NrMacSchedulerOfdma`, preservando latência do URLLC sem colapsar o *throughput* agregado [Villegas e Costa 2024].

Configuração dos três slices (exemplo reprodutível)

A título de referência (ajustável por cenário), utilizamos:

- eMBB: BWP de ≈ 20 MHz, $\mu \in \{1,2\}$; *On–Off* com pacotes médios (500 B) e taxas elevadas (10 pps);
- URLLC: BWP de $\approx$ 10–20 MHz, $\mu \in \{2,3\}$; pacotes pequenos (64 B), intervalos ativos curtos (20 pps) e *slots/mini-slots* reduzidos;
- mMTC: BWP de $\approx$ 5–10 MHz, $\mu = 0$; tráfego esparso (32 B, 2 pps) com foco em eficiência energética.

Esses valores mantêm coerência com o racional 3GPP/5G NR e permitem comparar, de forma controlada, o impacto de latência/*jitter*/PLR sobre a convergência do SL [Luna, Oliveira e Silva 2021, Koutlia e al. 2024, Medeiros, Santos e Almeida 2022]. Os parâmetros de cada *slice* a seguir estão alinhados às diretrizes de BWP/numerologia discutidas em seção 3.5.2 e à modelagem *On–Off* da seção 3.5.3. Para garantir reprodutibilidade, as variáveis de taxa (λ) e tamanho de pacote (`udpPacketSize`) são expostas no código-fonte e podem ser fixadas por cenário conforme o *rationale* 3GPP/5G NR citado. Adicionalmente, adotamos o padrão `MaxPackets` = 0*x*FFFFFFFF (envio "sem limite" durante o período de simulação), de modo que o efeito primário decorra da configuração de intervalo/*bitrate* e não de um contador finito de pacotes.

eMBB — configuração do gerador de tráfego. O Algoritmo 3.2 documenta a configuração do fluxo eMBB, cujo objetivo é sustentar taxas elevadas de dados com pacotes de tamanho moderado e intervalos compatíveis com $\mu \in \{1,2\}$. O parâmetro λ_{Be} regula o intervalo entre pacotes via `Interval` = `Seconds`$(1.0/\lambda_{Be})$. Tamanhos de pacote mais altos aumentam o *throughput* oferecido e, portanto, exercem maior pressão sobre a vazão útil do BWP mapeado ao eMBB (seção 3.5.3). Essa pressão é intencional: ela evidencia, nos resultados, a relação entre eficiência espectral e estabilidade do SL, uma vez que atrasos de fila no enlace impactam diretamente o tempo de ida/volta das ativações e gradientes.

Conforme ilustrado no Algoritmo 3.2, a configuração do tráfego eMBB é realizada por meio da classe `UdpClientHelper`, ajustando atributos de porta, tamanho de pacote e intervalo de envio de acordo com os requisitos de largura de banda e eficiência espectral.

Algoritmo 3.2 Trecho de código eMBB – tráfego de banda larga móvel aprimorada

```
1 UdpClientHelper dlClientVoice;
2 dlClientVoice.SetAttribute("RemotePort", UintegerValue(dlPortVoice));
3 dlClientVoice.SetAttribute("MaxPackets", UintegerValue(0xFFFFFFFF));
4 dlClientVoice.SetAttribute("PacketSize", UintegerValue(udpPacketSizeBe));
5 dlClientVoice.SetAttribute("Interval", TimeValue(Seconds(1.0/lambdaBe)));
```

Esse trecho de código reforça a estratégia metodológica adotada, ao transformar parâmetros de rede em variáveis explícitas que podem ser manipuladas em diferentes cenários de simulação.

Comentário técnico (eMBB). (i) `RemotePort` identifica a porta do fluxo eMBB, permitindo sua segregação nos filtros do `FlowMonitor`; (ii) `PacketSize` e λ_{Be} definem a oferta de carga: aumentos em ambos intensificam *throughput* e ocupação de recursos no BWP e eMBB; (iii) o uso de numerologias moderadas ($\mu = 1,2$) reduz a latência por *slot* em relação a $\mu = 0$, sem a agressividade de URLLC, compondo um cenário realista de "consumo pesado" porém não ultra-sensível a atraso.

URLLC — configuração para baixa latência/alta confiabilidade. O Algoritmo 3.3 apresenta o fluxo URLLC, ajustado com pacotes pequenos (`udpPacketSizeULL`) e intervalos curtos (λ_{ULL} elevado), coerente com $\mu \in \{2,3\}$. Essa combinação reduz o tempo de ocupação por pacote no enlace e, somada à granularidade temporal menor (*slots/mini-slots*), sustenta latências de ida/volta compatíveis com requisitos de controle ou aplicações sensíveis ao tempo. No contexto do SL, isso beneficia o *round-trip* de ativações/gradientes, mitigando violações de *deadline* por lotes.

O Algoritmo 3.3 detalha a parametrização do fluxo URLLC, evidenciando como a combinação de pacotes pequenos e intervalos curtos, em numerologias elevadas, sustenta baixa latência e alta confiabilidade.

Algoritmo 3.3 Trecho de código URLLC – tráfego ultraconfiável e de baixa latência

```
1 UdpClientHelper dlClientLowLat;
2 dlClientLowLat.SetAttribute("RemotePort", UintegerValue(dlPortLowLat));
3 dlClientLowLat.SetAttribute("MaxPackets", UintegerValue(0xFFFFFFFF));
4 dlClientLowLat.SetAttribute("PacketSize", UintegerValue(udpPacketSizeULL));
5 dlClientLowLat.SetAttribute("Interval", TimeValue(Seconds(1.0/lambdaULL)));
```

A implementação apresentada não apenas configura o tráfego, mas também evidencia o vínculo direto entre os atributos de rede (latência, *jitter*, vazão) e o comportamento do SL.

Comentário técnico (URLLC). (i) A escolha de tamanhos de pacote reduzidos limita variação de atraso (*queueing* e *serialization*); (ii) λ_{ULL} elevada produz maior cadência de amostras para a fila do BWP URLLC, mas a numerologia mais alta e a priorização no escalonamento (quando habilitada) acomodam tal cadência com menor *jitter*; (iii) esse desenho cria o contraste metodológico desejado com eMBB: latências inferiores, porém *throughput* agregado menor, permitindo observar o *trade-off* no impacto sobre o SL.

mMTC — atribuição de tipo de dispositivo e heterogeneidade. O Algoritmo 3.4 materializa a estratégia de heterogeneidade de dispositivos, classificando metade dos UEs como IoT (1) e metade como *smartphones* (0). Essa divisão 50/50 não é um fim em si, mas

um marco experimental para controlar a mistura de tráfegos esparsos (telemetria) e tráfegos mais intensivos, mantendo o BWP mMTC focado em eficiência energética ($\mu = 0$) e pacotes reduzidos. A separação *device-aware* facilita, na análise, relatórios estratificados (por tipo de dispositivo, *slice* eBWP), evidenciando como o perfil IoT contribui para padrões de PLR e economia energética diferentes daqueles de *smartphones*.

A lógica apresentada no Algoritmo 3.4 mostra como os dispositivos são classificados entre IoT e *smartphones*, permitindo simular a heterogeneidade típica do cenário mMTC e avaliar impactos diferenciados em PLR e consumo energético.

Algoritmo 3.4 Trecho de código mMTC – tráfego massivo do tipo máquina

Input: *totalUENum*

```
for i ← 0 to totalUENum − 1 do
    if i < totalUENum / 2 then
        deviceTypeVector[i] ← 1                    // IoT
    else
        deviceTypeVector[i] ← 0                    // Smartphone
```

Assim, o algoritmo funciona como elo entre a modelagem teórica (BWP/numerologia) e a instrumentação prática no ns-3, garantindo reprodutibilidade e clareza metodológica.

No caso do eMBB, a adoção de valores elevados de λ combinada a pacotes de tamanho médio exerce maior pressão sobre a vazão, o que torna evidentes os gargalos relacionados ao escalonamento e às filas de transmissão. Por sua vez, o URLLC mostra como o emprego de numerologias mais altas, associado ao envio de pacotes pequenos, contribui para a redução da latência observada no ciclo de SL. Essa redução, entretanto, ocorre ao custo de uma menor eficiência espectral, evidenciando um dilema entre desempenho em tempo real e aproveitamento de recursos. Já no contexto do mMTC, a presença de dispositivos heterogêneos — desde sensores de IoT até *smartphones* — ressalta o *trade-off* entre a robustez de cobertura e o consumo energético, apontando para a necessidade de políticas de alocação mais sensíveis ao perfil de cada tipo de usuário.

Scheduler customizado para priorização

Quando necessário, um escalonador customizado (`MyCustomScheduler`) ajusta pesos por *slice* para privilegiar tráfego sensível a atraso sem negligenciar vazão. A lógica se integra ao `NrMacSchedulerOfdma` e pode ser estendida para políticas com 5QI/*Data Radio Bearer* (DRB), preservando a comparabilidade experimental [Villegas e Costa 2024].

3.6 Modelo Único Compartilhado

Consideramos um único modelo de aprendizado compartilhado entre os UEs. As fatias de rede afetam apenas as condições de comunicação (latência, perdas, vazão), não havendo modelos separados por *slice*. Isso preserva comparabilidade entre cenários e isola o efeito de rede no processo de convergência do SL.

3.7 Arquitetura Integrada

A Figura 3.1 apresenta a arquitetura concebida para integrar o simulador 5G-LENA ao módulo de aprendizado por meio do ns3-ai. Nesse arranjo, cada cliente é responsável por processar as camadas iniciais do modelo de rede neural e, ao final dessa etapa, envia as ativações z ao servidor. Esse mecanismo de divisão das camadas tem como objetivo equilibrar a carga computacional entre os dispositivos de borda e o nó central, além de reduzir o volume de dados a serem transmitidos em comparação com o envio de amostras brutas. O fluxo de informações, portanto, inicia-se na periferia da rede, refletindo o caráter distribuído e colaborativo do aprendizado.

No lado do servidor, as ativações recebidas são utilizadas para prosseguir com as etapas subsequentes do modelo. O servidor executa a predição, calcula a função de perda $\mathcal{L}$ e, por meio do processo de retropropagação, obtém os gradientes correspondentes (g). Essa fase concentra a maior parte do esforço computacional, aproveitando os recursos de processamento mais robustos disponíveis no nó central. A escolha de manter essa parcela do modelo no servidor não é trivial: ela decorre de uma estratégia consciente de balanceamento entre eficiência de rede e capacidade de processamento, uma vez que a execução integral do modelo em dispositivos limitados seria impraticável em cenários reais.

Por fim, os gradientes calculados retornam aos clientes, que atualizam localmente os pesos das suas camadas. Esse movimento de ida e volta de dados entre clientes e servidor caracteriza a essência do SL, no qual há uma interdependência contínua entre os dois lados da arquitetura. Ao permitir que cada cliente mantenha parte do modelo e participe ativamente do processo de atualização, preserva-se não apenas a eficiência computacional, mas também aspectos de privacidade, uma vez que os dados originais permanecem nos dispositivos de origem.

Esse acoplamento estruturado, viabilizado pela integração com o ns3-ai, torna possível investigar, de forma sistemática e reprodutível, como as condições de rede em cenários B5G influenciam diretamente as curvas de aprendizado do SL. Métricas como latência, *jitter*, perdas e variações de vazão deixam de ser meros parâmetros de comunicação e passam a atuar como determinantes do ritmo e da qualidade da

convergência do modelo. A arquitetura integrada, portanto, não apenas conecta duas áreas tradicionalmente analisadas de maneira isolada — redes e aprendizado de máquina —, mas cria um campo de experimentação unificado, capaz de revelar interações profundas entre desempenho de rede e eficiência de treinamento.

3.7.1 Integração SL e B5G via ns3-ai

O Algoritmo 3.5 apresenta a lógica de orquestração entre o SL e o simulador ns-3 por meio do módulo ns3-ai. No SL, o treinamento do modelo é dividido entre dois agentes principais: o cliente, que processa as camadas iniciais sobre dados locais, e o servidor, responsável pelas camadas finais e pelo cálculo da função de perda, com devolução dos gradientes à camada de corte. Essa divisão reduz a carga computacional/energética do terminal e preserva a privacidade dos dados brutos, sendo particularmente adequada a cenários móveis heterogêneos e com restrições de largura de banda/latência, como em B5G.

Algoritmo 3.5 SplitLearning–NS3 (*ns3-ai*) – integração entre simulação de rede e aprendizado

```
/* Servidor (pesos w_s)                                                    */
1  Inicializar w_s
2  for cada rodada t = 1,2,...,N_r do
3      Receber z_i dos clientes via ns3-ai  ŷ_i ← f_s(z_i; w_s)
       L_i ← L(ŷ_i, y_i)  g_i ← ∂L_i/∂z_i
       Enviar g_i ao cliente i via ns3-ai
       Atualizar w_s por retropropagação

/* Integração ns-3 - ns3-ai                                                */
4  while loop de simulação do ns-3 do
5      Monitorar métricas de rede
       z_i ← ReceberDoCliente(i)
       g_i ← ObterGradienteServidor(z_i)
       Enviar g_i ao cliente i

/* Cliente i (pesos w_c^i)                                                 */
6  Receber configuração inicial w_c^i
7  for cada época local e = 1,...,E do
8      Particionar P_i em mini-lotes B_i
9      for cada lote b ∈ B_i do
10         z_i ← f_c(b; w_c^i)
           ▷ propagação até a cut layer
           Enviar z_i ao servidor via ns3-ai
           Receber g_i  w_c^i ← w_c^i − η · ∇_{w_c^i} L(z_i; g_i)
```

Seja $z_i = f_c(x_i; w_c^i)$ a saída até a *cut layer* no cliente (linha 10) e $\hat{y}_i = f_s(z_i; w_s)$ a predição no servidor (linha 3). A perda $\mathcal{L}_i = \mathcal{L}(\hat{y}_i, y_i)$ gera o gradiente $g_i = \partial\mathcal{L}_i/\partial z_i$ (linha 3) para a retropropagação local, enquanto o servidor atualiza w_s (linha 1). Essa organização mitiga *stragglers* e adequa-se a dispositivos de borda com recursos limitados, desde que a camada de corte seja escolhida de modo a balancear custo computacional local e volume de ativações transmitidas (ver discussão sobre *cut layer* na seção 2.2.3).

CAPÍTULO 4

Avaliação e Resultados

Este capítulo apresenta os resultados experimentais obtidos a partir da integração entre SL e o simulador ns-3/5G-LENA, considerando múltiplas métricas de rede em cenários B5G/6G. A análise é organizada em duas frentes: (i) o impacto direto das métricas de rede sobre a acurácia e o tempo de convergência do modelo; e (ii) análises comparativas entre *slices* e configurações de BWPs.

4.1 Avaliação e Discussões

Antes da análise individual de cada métrica, foi necessário validar a fidelidade do cenário. As distribuições de latência e vazão observadas mostraram-se consistentes com estudos anteriores conduzidos no 5G-LENA, assegurando a confiabilidade do ambiente experimental [CTTC 2025, Sousa 2022]. Além disso, verificou-se que a variabilidade entre usuários é tão ou mais relevante que os valores médios: variações de latência e *jitter* impactaram de forma significativa a estabilidade do treinamento, mesmo quando os valores médios se mantinham em níveis aparentemente aceitáveis. Esse resultado reforça que médias isoladas podem induzir a interpretações otimistas, mas é a análise da dispersão estatística que de fato revela a robustez do sistema em cenários distribuídos [Koutlia et al. 2023, Luna, Ferreira e Rocha 2021].

Outro aspecto fundamental foi a modelagem de tráfego *On–Off*, especialmente no *slice* mMTC, que introduziu padrões intermitentes e *bursty*, amplificando oscilações de *jitter* e ocasionando perdas pontuais. Essa fidelidade estatística aproxima os resultados das condições reais encontradas em redes IoT massivas, garantindo que as métricas formalizadas no Capítulo 3 (latência, vazão, PLR e energia) sejam interpretadas de forma consistente e alinhadas ao comportamento esperado em cenários práticos.

Para orientar a leitura do restante desta seção, apresenta-se a seguir uma síntese conceitual das relações entre as métricas de rede e os efeitos observados no SL. Essa descrição não possui caráter numérico, mas fornece uma visão de alto nível que serve de guia para compreender como cada parâmetro impacta o processo de treinamento distribuído.

Os resultados quantitativos, acompanhados dos intervalos de confiança, encontram-se detalhados nas figuras específicas de cada métrica e, de forma consolidada, na Tabela 4.1.

A primeira métrica considerada é a latência. Quando os valores de latência se elevam, o tempo de convergência do modelo ao longo das iterações tende a se alongar de maneira perceptível. Esse atraso prolonga cada ciclo de comunicação entre cliente e servidor, fazendo com que o processo de aprendizado como um todo se torne mais lento. É importante destacar, contudo, que em situações em que a confiabilidade do enlace se mantém estável — ou seja, quando não há perdas significativas de pacotes —, a acurácia final do modelo não necessariamente sofre deterioração. Nesse caso, o impacto da latência se manifesta sobretudo no tempo necessário para alcançar um mesmo patamar de desempenho, mas não na qualidade intrínseca do resultado final.

A segunda métrica relevante é a PLR. Diferentemente da latência, cujo efeito pode ser atenuado pela robustez do algoritmo de treinamento, a PLR apresenta impacto direto e imediato sobre a acurácia do modelo. A perda de ativações ou gradientes transmitidos entre os nós reduz de forma irreversível a quantidade de informação útil disponível para atualização dos parâmetros, comprometendo a convergência e levando a quedas expressivas de desempenho. Por essa razão, a PLR é considerada o fator mais crítico entre as métricas de rede, uma vez que atua diretamente na essência do processo de aprendizado.

O *jitter*, definido como a variação temporal da latência, também exerce influência significativa sobre o SL. Ainda que os valores médios de atraso estejam sob controle, a presença de flutuações constantes compromete a previsibilidade do treinamento, gerando instabilidade na curva de aprendizado. Essa irregularidade temporal se traduz em ciclos inconsistentes, nos quais algumas iterações são concluídas rapidamente enquanto outras sofrem atrasos inesperados. O resultado é uma maior dificuldade em manter o ritmo de atualização do modelo e, consequentemente, uma aprendizagem menos estável ao longo do tempo. Outro parâmetro frequentemente observado é a vazão. Quando esse valor aumenta, o sistema consegue acelerar a transmissão de pacotes, reduzindo o tempo total necessário para completar uma rodada de treinamento. Isso se traduz em ganhos práticos na eficiência temporal, embora o impacto sobre a acurácia final seja geralmente moderado. Em outras palavras, a vazão elevada contribui para que o aprendizado aconteça em menos tempo, mas não garante, por si só, melhorias substanciais na qualidade do modelo obtido.

Finalmente, o consumo de energia merece destaque. Valores mais altos de gasto energético afetam sobretudo a viabilidade prática da execução do treinamento em dispositivos com recursos limitados, como sensores IoT ou *smartphones* com restrições de bateria. Ainda que a energia não interfira de forma direta na acurácia, o aumento do custo de execução pode tornar inviável a continuidade do processo em determinados contextos. Assim, o impacto se dá de forma indireta, pois limita a escala, a frequência ou mesmo a possibilidade de participação de alguns dispositivos no processo colaborativo de

aprendizado.

Essa análise conceitual reforça a ideia de que cada métrica de rede, ao se modificar, acarreta efeitos específicos e, em muitos casos, complementares sobre o SL. Ao compreender essas relações, torna-se possível interpretar com maior clareza os resultados experimentais que serão apresentados nas seções subsequentes, atribuindo a cada variação observada uma explicação fundamentada na dinâmica da rede e em sua interação com o processo de aprendizado distribuído.

4.1.1 Impacto dos Parâmetros de Rede

4.1.1.1 *Delay* (Latência)

A latência influencia diretamente o tempo de convergência do SL, já que cada iteração exige a transmissão de ativações ao servidor e o retorno de gradientes. Conforme as Equações (3-1) e (3-2), a latência foi avaliada pela média $\bar{x}$ e pelo desvio-padrão s dos atrasos por pacote.

Os experimentos realizados mostraram valores médios de latência na faixa de 6,3 ms a 7,5 ms, com acurácia de validação variando entre 0,932 e 0,960 (Figura 4.1). Esses resultados confirmam a hipótese teórica apresentada na Figura 2.3: aumentos moderados de atrasos tendem a alongar o tempo de convergência, mas não comprometem a acurácia final quando a confiabilidade do enlace é mantida.

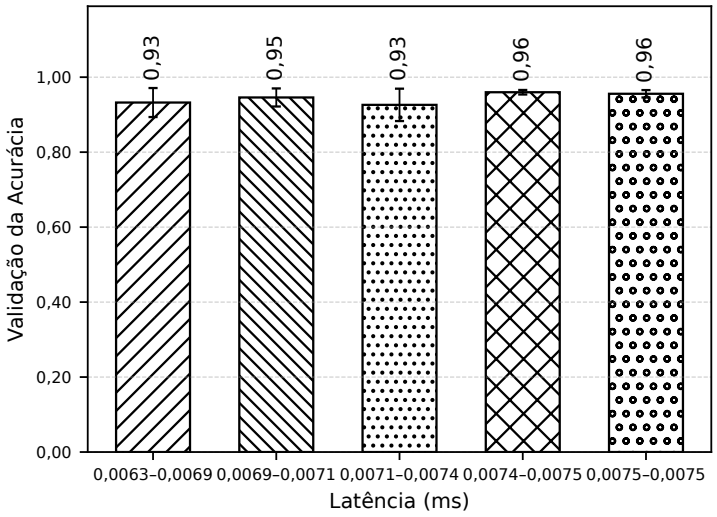

Figura 4.1: Latência em relação à acurácia de validação: maiores atrasos tornam a convergência mais lenta.

Contudo, quando a variabilidade da latência (desvio-padrão s) foi elevada, a curva de aprendizado tornou-se irregular, indicando que a dispersão temporal tem maior peso do que o valor médio isolado. Esse resultado reforça a relevância de políticas de alocação

e mecanismos de QoS para mitigar desequilíbrios no ciclo cliente–servidor, assegurando maior estabilidade no treinamento distribuído.

4.1.1.2 *Throughput* (Vazão)

A vazão, calculada de acordo com a Equação (3-3), representa a capacidade efetiva da rede em sustentar transmissões paralelas de ativações (*uplink*) e gradientes (*downlink*). Nos experimentos, os valores médios de vazão oscilaram entre aproximadamente 0,0028 Mbps e 0,33 Mbps, com acurácia de validação variando de 0,913 a 0,957 (Figura 4.2). Observa-se que maiores taxas de vazão reduzem o tempo de iteração e aceleram a convergência, confirmando a expectativa teórica ilustrada na Figura 2.4.

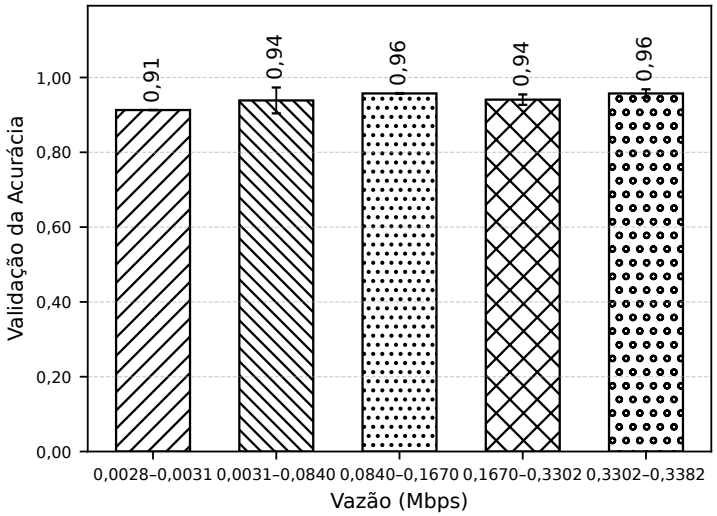

Figura 4.2: Relação entre vazão e acurácia - maior vazão acelera o treino, mas impacto limitado na acurácia final.

Quando a vazão foi artificialmente limitada, o impacto direto sobre a acurácia final foi moderado, mas o tempo de treinamento aumentou de forma significativa. Esse resultado evidencia que a vazão exerce influência principalmente na eficiência temporal do processo de treinamento, embora não se mostre tão determinante para a acurácia quanto a taxa de perda de pacotes (PLR). Assim, assegurar largura de banda adequada é importante, mas insuficiente se não houver confiabilidade na entrega dos pacotes.

4.1.1.3 *Jitter*

O *jitter*, entendido como a variabilidade da latência, também está associado às Equações (3-1) e (3-2). Nos experimentos, os valores observados oscilaram de 0,0001 s a 0,0026 s, com acurácia de validação entre 0,931 e 0,957 (Figura 4.3). Apesar dos valores médios baixos, períodos de *jitter* elevado resultaram em oscilações na curva de

aprendizado, introduzindo instabilidades na convergência. Esse efeito foi articularmente evidente no *slice* mMTC, onde o tráfego intermitente modelado pelo padrão *On–Off* acentuou a irregularidade temporal.

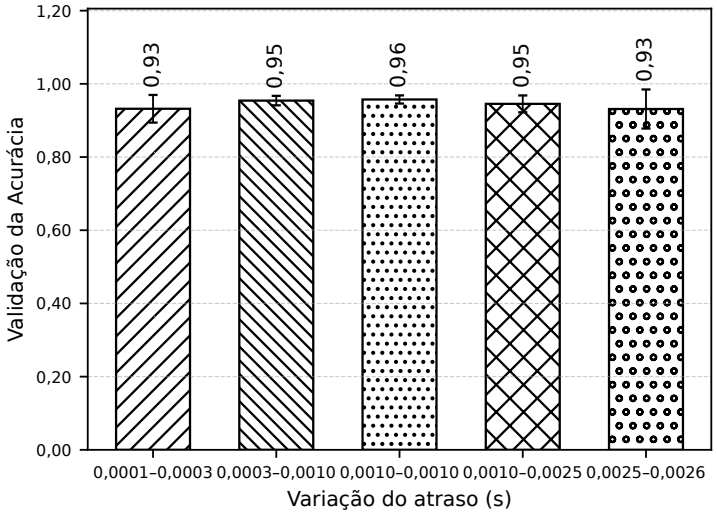

Figura 4.3: *Jitter* comparada à acurácia - flutuações comprometem a estabilidade do SL, sobretudo em URLLC.

Assim, o *jitter* não reduz de forma imediata a acurácia final, mas compromete a previsibilidade e a suavidade da curva de aprendizado. Esse aspecto é crítico em aplicações sensíveis a atrasos, como serviços de URLLC, nas quais oscilações temporais podem inviabilizar a operação mesmo quando a acurácia média final é mantida.

4.1.1.4 Taxa de perda de pacotes (PLR)

A PLR definida pela Equação (3-4), mostrou-se a métrica mais crítica entre todas as avaliadas. Nos experimentos, a PLR variou de 0,0 a 13,3%, com impacto direto na acurácia de validação, que caiu de 0,950 para 0,919 (Figura 4.4). Diferentemente da latência ou da vazão, que afetam principalmente o tempo de convergência, a PLR compromete a qualidade do aprendizado de forma irreversível, pois o servidor deixa de receber parte significativa dos gradientes necessários à atualização do modelo.

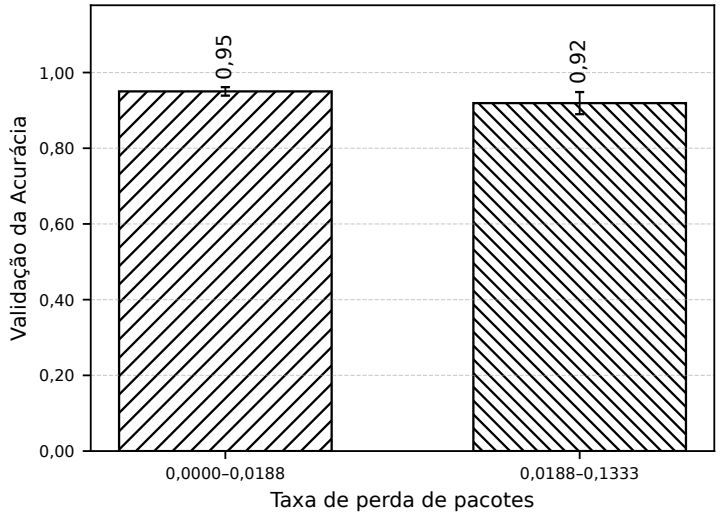

Figura 4.4: LR frente à acurácia - perdas de pacotes reduzem diretamente a acurácia final.

Esse resultado confirma a hipótese central desta dissertação: a confiabilidade do enlace, refletida pela PLR, é o fator determinante para a viabilidade do SL em redes B5G/6G. A redução de perdas deve, portanto, ser prioridade em arquiteturas de aprendizado distribuído que visam equilíbrio entre desempenho e robustez.

4.1.1.5 Consumo de Energia

O consumo energético, definido pela Equação (3-5), foi avaliado como métrica complementar. Nos cenários simulados, os valores variaram entre 0,2085 J e 0,2329 J, com acurácia de validação entre 0,913 e 0,957 (Figura 4.5). Observou-se que UEs associados ao *slice* eMBB apresentaram maior gasto energético, reflexo da maior potência de transmissão e do tráfego contínuo. Já o *slice* mMTC mostrou consumo reduzido, compatível com seu perfil intermitente.

Tabela 4.1: Impacto das métricas de rede sobre a acurácia do SL. PLR se destaca como métrica crítica para convergência.

Parâmetro	**Tendência → Acurácia**	**Impacto Máx. / Cenário**	**Impacto Médio (%)**	**Cenário Crítico**
Energia (J)	Maior → maior	+13,4% / URLLC-2	−2,0%	eMBB/1
jitter (s)	Maior → menor	−23,6% / URLLC-2	−23,9%	eMBB/1
Latência (s)	Maior → maior	+57,6% / URLLC-2	+4,2%	URLLC/2
PLR (%)	Maior → menor	−70,1% / eMBB-1	−52,2%	eMBB/1
Vazão (Mbps)	Maior → maior	+12,2% / mMTC-0	+10,2%	URLLC/2

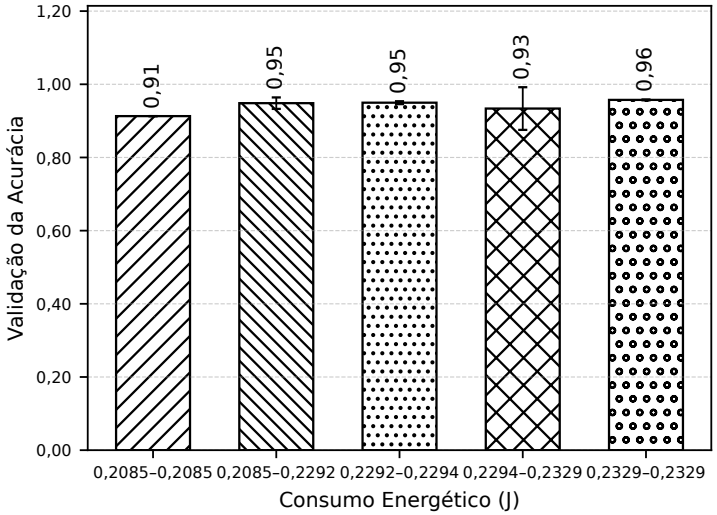

Figura 4.5: Consumo energético em relação à acurácia: custo prático maior, sem efeito direto na acurácia.

Embora o consumo energético não afete diretamente a acurácia final, ele define a viabilidade prática do SL em dispositivos restritos, como IoT. Existe, portanto, um *trade-off* claro: maior potência de transmissão reduz a latência e melhora a eficiência do treinamento, mas eleva o custo energético. É importante destacar que o impacto sobre a acurácia é indireto: o ganho ocorre pela redução do tempo de convergência e não por uma alteração intrínseca na qualidade do modelo. Esse resultado reforça a importância de estratégias de escalonamento conscientes de energia, especialmente em ambientes heterogêneos.

Síntese multimétrica por *slice*. Após a análise individual de latência, vazão, *jitter*, PLR e energia, apresentamos a Figura 4.6, que consolida, em uma única visualização, o perfil comparativo dos *slices* URLLC, eMBB e mMTC. O objetivo é evidenciar, de forma integrada, os *trade-offs* característicos de cada perfil e como esses compromissos dialogam com os requisitos do SL em redes B5G/6G.

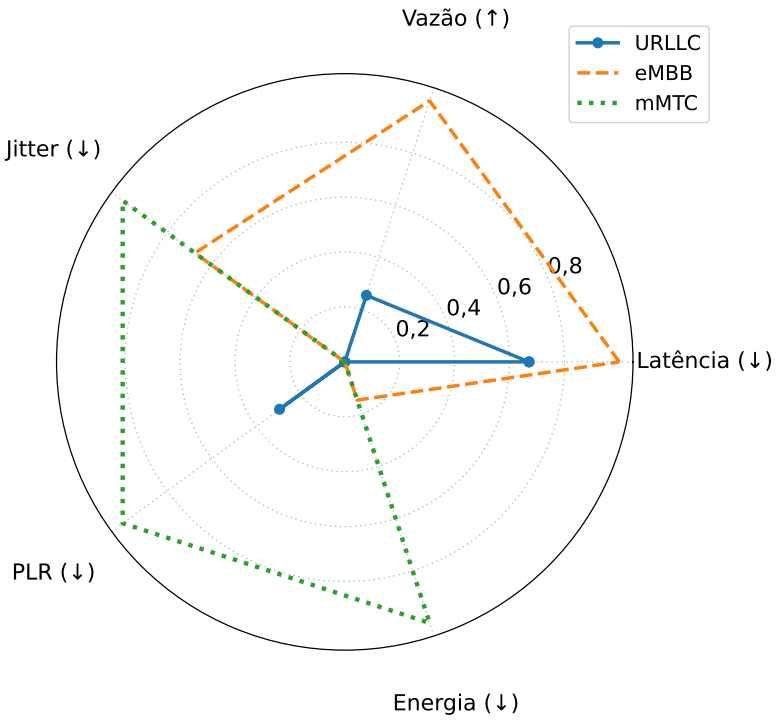

Figura 4.6: Radar multimétrico: URLLC prioriza latência, eMBB vazão, mMTC eficiência energética.

Interpretação e achados que trazem robustez. A leitura da Figura 4.6 confirma e integra as evidências empíricas apresentadas nas subseções anteriores. O URLLC evidencia-se pela previsibilidade temporal e pela confiabilidade, concentrando-se nos eixos de latência reduzida e PLR mínima, com valores contidos de *jitter*, o que o alinha às exigências de aplicações críticas em tempo quase real e sustenta uma convergência estável do SL mesmo sob janelas de comunicação curtas. Para o eMBB revela dominância em vazão, o que viabiliza ciclos de treinamento mais rápidos; todavia, esse ganho é acompanhado de maior custo energético, compatível com tráfego contínuo e potência de transmissão elevada. A acurácia nesse perfil mantém-se elevada quando a PLR permanece baixa, reforçando que capacidade sem confiabilidade não é suficiente. Por sua vez, o mMTC destaca-se pela eficiência energética, fator central para dispositivos IoT, mas mostra maior vulnerabilidade a oscilações de *jitter* e incrementos de PLR, especialmente em razão do padrão de tráfego *On–Off*, o que pode comprometer a estabilidade do aprendizado em cenários densos.

Em conjunto, esses resultados reiteram o papel transversal da PLR como métrica mais crítica para a acurácia do SL, enquanto latência e vazão modulam predominantemente o tempo de convergência. O radar, portanto, não apenas sintetiza tendências, mas materializa os compromissos de projeto, ao mostrar que o URLLC prioriza confiabilidade e previsibilidade, o eMBB oferece alta capacidade a custo energético, e o mMTC viabiliza

escala com baixo consumo, exigindo mitigação ativa da variabilidade temporal.

4.1.1.6 Sobrecarga de Controle

Embora esta dissertação tenha se concentrado nas métricas diretamente associadas ao plano de dados (*User Plane* (UP)), é importante reconhecer que o desempenho observado em redes reais também sofre influência significativa do plano de controle (*Control Plane* (CP)). Mensagens de sinalização RRC, *Non-Access Stratum* (NAS) e procedimentos de *handover* adicionam *overhead* que pode aumentar a latência fim a fim e afetar a estabilidade do SL. Os resultados aqui reportados refletem um ambiente de simulação em que a sobrecarga de sinalização foi abstraída. Em cenários operacionais, a mobilidade dos usuários, os eventos de *handover* e as interações de CP podem introduzir atrasos adicionais e variabilidade, especialmente no caso do URLLC. Dessa forma, os valores obtidos nesta avaliação devem ser interpretados como limites de referência sob condições ideais, úteis para comparações controladas, mas não como garantias absolutas de desempenho em campo.

No ambiente de simulação adotado, tais efeitos foram abstraídos, de forma a isolar a análise das métricas centrais (latência, vazão, *jitter*, PLR e energia). Entretanto, em cenários reais, o custo de sinalização pode introduzir variações adicionais de atraso e consumo energético, sobretudo em aplicações URLLC. Essa limitação deve ser levada em conta ao extrapolar os resultados para ambientes operacionais.

4.1.2 Análises Complementares *Slice*/BWP

As métricas formalizadas no Capítulo 3 e analisadas individualmente em seção 4.1.1 também foram avaliadas sob a perspectiva de *slices* e BWPs. Essa visão cruzada permite compreender como diferentes perfis de tráfego influenciam simultaneamente latência, vazão, PLR e consumo energético.

A análise por *slice* evidenciou padrões consistentes com as diretrizes do 3GPP. O perfil URLLC apresentou latências estáveis e reduzidas, praticamente isentas de perdas, além de elevada previsibilidade, o que confirma sua adequação para aplicações críticas que exigem alta confiabilidade. O perfil eMBB concentrou mais de 60% da vazão total observada, assumindo papel essencial em aplicações multimídia e de grande demanda de dados, embora esse desempenho esteja associado a maior consumo energético. Por sua vez, o perfil mMTC caracterizou-se por tráfego esparso e baixo consumo energético, em conformidade com os requisitos típicos de dispositivos de IoT, mas também revelou maior suscetibilidade a perdas em virtude da natureza intermitente de seu tráfego.

É importante destacar que tais resultados refletem condições controladas de simulação, nas quais fatores como mobilidade, sobrecarga de sinalização e procedimentos

de *handover* foram abstraídos. Em redes reais, esses elementos introduzem atrasos adicionais e maior variabilidade, reduzindo a previsibilidade observada, sobretudo em cenários críticos como o URLLC. Dessa forma, os valores reportados devem ser entendidos como limites de referência sob hipóteses ideais, úteis para comparações entre *slices*, mas não como garantias absolutas em ambientes operacionais.

No que se refere à comparação entre diferentes configurações de BWPs, verificou-se que numerologias mais altas favoreceram a redução da latência, mas essa melhoria ocorreu às custas de menor robustez e de uma cobertura reduzida. Em contrapartida, numerologias mais baixas proporcionaram maior confiabilidade em cenários de longas distâncias, ainda que com tempos de transmissão mais elevados. Esse comportamento reforça a existência de um compromisso inerente entre latência, robustez e alcance, o qual deve ser considerado de forma criteriosa no dimensionamento de sistemas B5G/6G baseados em *network slicing*.

Como visão de projeto, a Figura 4.7 resume, de forma qualitativa (sem números), os comportamentos esperados por *slice* e seus efeitos típicos sobre o SL. Em seguida, confrontamos essa expectativa com os resultados quantitativos desta subseção e da Tabela 4.2.

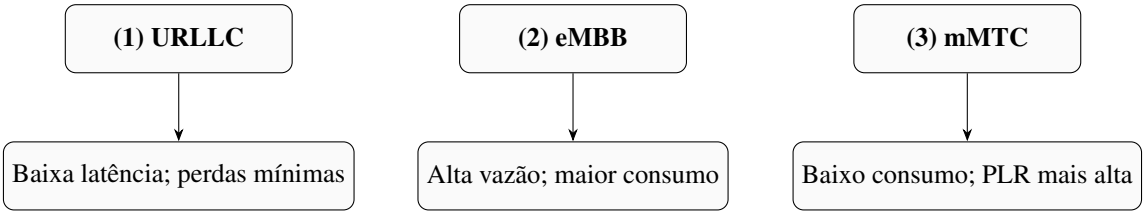

Figura 4.7: Perfis de *slice*: cada perfil prioriza uma métrica crítica, moldando a resiliência do SL.

A leitura da Figura 4.7 antecipa os achados por *slice*: URLLC prioriza previsibilidade temporal, eMBB prioriza vazão (a custo de energia), e mMTC prioriza baixo consumo, porém com maior risco de irregularidades temporais. A Tabela 4.2 consolida esses efeitos com base nos resultados obtidos.

Em suma, a leitura cruzada por *slice*/BWP confirma os padrões da Figura 4.7 e prepara as recomendações consolidadas na Síntese.

4.2 Resumo dos Resultados

A síntese dos resultados deste capítulo pode ser compreendida a partir da articulação entre os diferentes *slices* e as métricas que mais influenciam o desempenho do SL. Embora originalmente fosse apresentada em forma de diagrama, a mesma lógica pode

Tabela 4.2: Resumo comparativo das métricas de rede nos cenários simulados

Métrica	**Tendência observada**	**Perfil favorecido**	**Observação**
Energia (J)	Heterogênea por perfil de tráfego	mMTC (baixo), eMBB (alto)	Relevante para viabilidade IoT massiva.
jitter (s)	Variação moderada na média	eMBB	Maior instabilidade no mMTC (padrão On–Off).
Latência (s)	Baixa e estável nos perfis críticos	URLLC	Maior variabilidade no mMTC (tráfego intermitente).
PLR (%)	Baixa nos cenários *baseline*; sob estresse, degrada fortemente	–	Quando elevada, domina a queda de acurácia.
Vazão (Mbps)	Taxa dominante em cenários multimídia	eMBB	URLLC estável; mMTC residual.

ser explicitada de maneira narrativa, permitindo compreender de forma clara as relações que sustentam a análise.

No caso do *slice* URLLC, observou-se que a latência reduzida e a baixa taxa de perdas de pacotes constituem fatores dominantes. Esses dois elementos atuam de forma complementar para garantir uma convergência estável e rápida do processo de aprendizado, sem comprometer a acurácia final. Trata-se de um perfil de tráfego que, por sua natureza crítica, prioriza a confiabilidade do enlace, de modo que a preservação da acurácia está diretamente associada ao controle rigoroso da latência e do *packet loss*.

O perfil eMBB, por sua vez, está fortemente associado à elevada taxa de vazão e à necessidade de manter as perdas de pacotes em níveis reduzidos. Essa combinação viabiliza ciclos de treinamento mais ágeis, uma vez que o fluxo de dados é mais intenso e contínuo. Contudo, diferentemente do URLLC, a acurácia do modelo nesse cenário revela maior sensibilidade às oscilações de confiabilidade, indicando que o ganho em rapidez deve vir acompanhado de garantias adicionais de estabilidade para que os resultados se mantenham consistentes.

No contexto do mMTC, a métrica de energia adquire papel central, refletindo a necessidade de assegurar eficiência em dispositivos com recursos limitados. Embora esse perfil apresente como benefício a alta economia energética, a elevação da taxa de perdas de pacotes desponta como fator de risco, capaz de comprometer a estabilidade do processo de aprendizado e provocar redução significativa na acurácia. Em outras palavras, o mMTC expõe um dilema típico: ampliar a viabilidade prática em dispositivos IoT por meio da eficiência energética implica a adoção de estratégias complementares para mitigar a vulnerabilidade associada às perdas.

Essas relações não ocorrem de forma isolada. Os efeitos cruzados entre métricas revelam que a confiabilidade — representada pela PLR — exerce impacto transversal sobre todos os perfis. Pequenas elevações nessa métrica repercutem diretamente na acurácia, o que a torna mais crítica do que latência, vazão ou energia quando analisada em conjunto.

Nesse sentido, a Tabela 4.3 organiza quantitativamente esses achados, permitindo comparar de maneira estruturada o peso relativo de cada métrica e as recomendações derivadas.

Tabela 4.3: Resumo dos impactos das métricas de rede sobre o SL

Métrica	**Impacto sobre o SL**	**Recomendação**
Energia (J)	Define viabilidade prática em IoT	Balancear consumo e desempenho com escalonamento eficiente
Jitter (s)	Instabilidade na curva de aprendizado	Políticas de QoS para suavizar variações
Latência (s)	Atraso na convergência, mas acurácia preservada	Usar numerologias mais altas em cenários críticos
PLR (%)	Queda direta e crítica na acurácia	Minimizar perdas via alocação robusta e redundância
Vazão (Mbps)	Acelera ciclos de treino, efeito moderado na acurácia	Priorizar *slices* eMBB em alta demanda
Síntese Geral	PLR foi a métrica mais crítica, seguida por latência e vazão	Projetar SL priorizando confiabilidade do enlace

A integração entre a análise qualitativa e a sistematização quantitativa evidencia que a PLR constitui a métrica mais crítica para a convergência do SL, seguida por latência e vazão. O *jitter* e o consumo energético, embora secundários, adquirem relevância em cenários específicos, como no tráfego massivo de IoT. Dessa forma, consolida-se a contribuição central desta dissertação: projetar arquiteturas de aprendizado distribuído em redes B5G/6G requer uma abordagem que privilegie a confiabilidade do enlace, sem desconsiderar a eficiência energética e a estabilidade do processo de aprendizado.

4.3 Validação com Mundo Real

Embora os experimentos tenham sido conduzidos integralmente em ambiente de simulação, os resultados obtidos apresentam consonância com medições reportadas em *testbeds* e ensaios de campo em 5G/5G NR [Lagén et al. 2023, Koutlia et al. 2023]. Em particular, as latências médias observadas nos cenários URLLC (ordem de 6–7 ms; ver Figura 4.1) situam-se na faixa tipicamente reportada para enlaces configurados com numerologias elevadas e *mini-slots*, bem como com mecanismos de agendamento voltados à confiabilidade [Park et al. 2022, Alfadhli et al. 2019, Larrañaga et al. 2023]. Vale acrescentar ainda, a PLR variando de 0% a 13,3% (Figura 4.4) é compatível com regimes de

carga sob interferência e competição de recursos em células densas, conforme documentado em avaliações URLLC e de *network slicing* [Amjad et al. 2021, Popovski et al. 2018, Khan et al. 2022]. Finalmente, a sensibilidade da curva de aprendizado ao *jitter* (Figura 4.3) está alinhada a estudos que apontam a variabilidade temporal do enlace como fator crítico para controle e inferência em tempo quase real, especialmente em arquiteturas de *split/edge* sobre redes sem fio [Itahara, Nishio e Yamamoto 2021, Lin et al. 2024, Liu, Deng e Mahmoodi 2023].

Ressalte-se que o ambiente adotado abstrai parte do *overhead* de sinalização do CP (Seção 4.1.1.6); portanto, as latências fim a fim em redes operacionais tendem a incluir parcelas adicionais de atraso e variabilidade. Ainda assim, as tendências causais identificadas neste capítulo permanecem: (i) a PLR domina a acurácia final do SL; (ii) a latência média e a vazão modulam o tempo de convergência; e (iii) o *jitter* afeta a estabilidade do treinamento, especialmente sob tráfego intermitente (mMTC). Esses achados são coerentes com a literatura recente sobre aprendizado distribuído *over-the-air* e serviços sensíveis a atraso em 5G/B5G (p. ex., estudos de *testbed* e campanhas de medição em ambientes urbanos densos).

Em síntese, apesar das simplificações inerentes ao simulador, a comparação qualitativa com resultados empíricos sustenta a validade externa das conclusões e reforça a utilidade dos cenários propostos como *proxy* para redes operacionais. Como passo subsequente, recomenda-se a replicação parcial dos cenários em *testbeds* físicos para calibração fina de parâmetros e verificação de limites práticos de vazão, PLR e latência.

4.4 Implicações Práticas em Setores Críticos

Como antecipado na Introdução (Capítulo 1), setores como saúde, transporte autônomo e Indústria 4.0 foram destacados como motivadores práticos deste estudo. Os resultados experimentais agora permitem detalhar como cada métrica de rede impacta diretamente esses cenários, em consonância com as previsões do 5G discutidas na literatura. De acordo com [Kurose e Ross 2020], a 5G foi concebida para sustentar aplicações críticas, como *Augmented Reality* (AR)/*Virtual Reality* (VR), veículos autônomos, robótica industrial e o *Fixed Wireless Access* (FWA). Tais aplicações ilustram a diversidade de requisitos de latência, confiabilidade e capacidade de conexão massiva, aspectos refletidos nos experimentos desta dissertação.

Os resultados obtidos não se restringem ao plano teórico, mas indicam implicações práticas em setores críticos, nos quais a confiabilidade da comunicação e a eficiência energética se configuram como fatores decisivos para a adoção de arquiteturas de SL. Ao demonstrar que a PLR tende a atuar como métrica dominante, enquanto latência e

vazão exercem influência moderada, esta pesquisa aproxima-se dos desafios concretos enfrentados em aplicações sensíveis.

4.4.1 Saúde

Aplicações médicas, como monitoramento remoto de sinais vitais e diagnósticos assistidos por AI, exigem baixa latência e perdas próximas de zero. Nossos experimentos mostraram que a PLR é a métrica dominante, com quedas acentuadas de acurácia quando acima de 10%. Nesse contexto, a preservação da confiabilidade do enlace (PLR $\downarrow$) torna-se requisito de segurança clínica: uma transmissão interrompida pode comprometer decisões médicas críticas.

4.4.2 Veículos Autônomos

Sistemas de condução autônoma dependem de respostas em milissegundos. Resultados de latência (6–7 ms em média, Figura 4.1) confirmam que atrasos moderados alongam o tempo de convergência, mas a acurácia pode ser preservada. Contudo, aumentos de *jitter* (Figura 4.3) introduzem instabilidades perigosas na curva de aprendizado. Portanto, políticas de alocação que minimizem variação temporal são essenciais para viabilizar o uso de SL em cenários de transporte autônomo.

4.4.3 Indústria 4.0

Em linhas de produção inteligentes, falhas de sincronização podem paralisar sistemas robóticos inteiros. Nossos achados confirmam que o eMBB concentra vazão, mas a um custo energético elevado (Figura 4.5). Assim, estratégias de escalonamento energético devem equilibrar eficiência e previsibilidade, garantindo que dispositivos IoT (mMTC) operem com baixo consumo sem sacrificar a confiabilidade global.

4.4.4 Mapa comparativo

A Tabela 4.4 sintetiza a relação entre métricas críticas de rede, impacto sobre o SL e setores mais sensíveis.

Tabela 4.4: Mapa de implicações práticas: métricas → impacto no SL → setor crítico

Métrica	**Impacto no SL**	**Setor crítico afetado**
Energia (J)	Define a viabilidade prática em dispositivos IoT	Indústria 4.0, IoT em larga escala
Jitter (s)	Instabilidade na curva de aprendizado	Transporte autônomo, URLLC em saúde
Latência (s)	Alongamento do tempo de convergência	Veículos autônomos (decisão em tempo real)
PLR (%)	Queda direta e crítica na acurácia	Saúde (monitoramento remoto, diagnósticos)
Vazão (Mbps)	Acelera ciclos de treino, mas efeito moderado na acurácia	Indústria 4.0 (robôs colaborativos, automação)

Em consonância com essas observações, é importante destacar que as aplicações críticas previstas para o 5G incluem AR/VR, veículos autônomos, robótica industrial e o FWA [Kurose e Ross 2020]. Esses domínios reforçam a diversidade de requisitos que justificam a ênfase nas métricas aqui avaliadas: confiabilidade de enlace para a saúde, latência ultrabaixa para o transporte autônomo e eficiência energética para a Indústria 4.0. Dessa forma, os resultados obtidos nesta dissertação não apenas validam cenários projetados na literatura, mas também evidenciam como o SL pode ser explorado como solução prática em setores críticos já delineados nas metas do 5G.

Essas implicações práticas reforçam a necessidade de arquiteturas resilientes, capazes de manter desempenho consistente sob condições realistas. A análise quantitativa desta dissertação fornece subsídios não apenas para a pesquisa acadêmica, mas também para a adoção de soluções baseadas em SL em setores de missão crítica. No próximo capítulo, essas evidências são retomadas e consolidadas nas conclusões e perspectivas futuras.

CAPÍTULO 5

Considerações Finais e Trabalhos Futuros

Este capítulo apresenta uma síntese das contribuições desta dissertação, destacando os principais resultados alcançados e sugerindo direções para pesquisas futuras.

5.1 Resultados Obtidos

Este item consolida a produção acadêmica, técnica e de *software* decorrente desta dissertação, em linha com o modelo de apresentação adotado em trabalhos do INF/UFG. A lista considera: (i) publicações/submissões diretamente ligadas ao tema central (arquitetura SL orientada a métricas de rede, integração ns-3/5G-LENA via ns3-ai);(ii) *software*/aplicativo associado ao *framework* experimental utilizado no LABORA/INF-UFG; e (iii) outras publicações/submissões.

Tabela 5.1: Publicação (alinhada ao tema).

Nº	**Ano**	**Produção**	**Tipo**	**Resultado**
1	2025	Reis, C. B., Ribeiro, M. R., Moreira, W. e Oliveira-Jr, A. SLArch: A Network Metric-aware Split Learning Architecture for B5G/6G Mobile Networks.	Conferência	Publicado na conferência IEEE 13th Wireless Days 2025

Tabela 5.2: *Software* e Repositório de Código. [Criado pelo Autor]

Nº	**Ano**	**Produção**	**Tipo**	**Resultado**
1	2025	Reis, C. B.	Repositório	https://github.com/LABORA-INF-UFG/SplitLearning-B5G

Tabela 5.3: Outras Publicações.

Nº	Ano	Produção	Tipo	Documento
1	2024	Silva, R.S, Oliveira, R. R., Carvalho, L., Freitas, L., Xavier, P., Reis, C. B., Oliveira-Jr, A. e Cardoso, K. V. Soluções baseadas em aprendizado por reforço profundo para implantar VANTs como gateways LoRaWAN com foco na Qualidade de Serviço de IoT. Anais do XLII SBrT, DOI: 10.14209/sbrt.2024.1571036460.	Conferência	https://biblioteca.sbrt.org.br/articles/4668

5.2 Conclusões

Este trabalho desenvolveu e avaliou a arquitetura *SplitLearning-ns3* [LABORA-INF/UFG 2025], integrando o paradigma de SL ao simulador ns-3/5G-LENA via ns3-ai, em um *framework* integrado, reprodutível e extensível. A proposta vai além de uma adaptação técnica: oferece um ambiente que conecta, de forma sistemática, métricas de desempenho de rede ao comportamento do aprendizado distribuído em cenários representativos de B5G/6G. Com isso, estabeleceu-se um elo claro entre as camadas de comunicação e de inteligência artificial, permitindo análises comparativas e replicáveis.

Os experimentos confirmaram a PLR como a métrica mais crítica para a estabilidade e a acurácia do SL. Quando as perdas superam 5%, observa-se comprometimento da convergência e atrasos no processo de aprendizado, chegando, em alguns casos, a resultados inconsistentes. Essa evidência reforça que a confiabilidade do enlace é requisito fundamental para a viabilidade prática do SL.

Latência e vazão apresentaram impacto moderado: não alteraram de forma decisiva a acurácia final, mas influenciaram o tempo para convergência e a eficiência global do treino. Maior latência demanda ciclos adicionais de interação cliente–servidor; restrições de vazão limitam o volume de dados por rodada, reduzindo a velocidade do processo. Embora secundárias em relação à PLR, tais métricas são estratégicas para o equilíbrio entre qualidade e tempo de execução.

Outros fatores, como *jitter* e consumo energético, mostraram relevância contextual. Em cenários mMTC, a variabilidade temporal compromete a regularidade do ciclo de aprendizado; contudo energia, ainda que menos diretamente associada à acurácia, torna-se determinante para a viabilidade em larga escala em dispositivos com recursos restritos.

Esses indicadores não podem ser ignorados diante da heterogeneidade das redes e da crescente presença de dispositivos IoT.

A análise por *slices* e BWPs corroborou padrões esperados pelas especificações do 3GPP: URLLC com latência baixa e perdas mínimas para aplicações críticas; eMBB com maior vazão para tráfego multimídia; e mMTC com comportamento intermitente e melhor eficiência energética. Esses resultados atestam a fidelidade do *framework* às premissas normativas e à operação prática dos perfis de serviço.

Metodologicamente, a avaliação de latência considerou não apenas valores médios, mas também a variabilidade estatística (Equações (3-1) e (3-2)). Verificou-se que altas taxas de perda combinadas a elevada dispersão dos atrasos degradam ainda mais a robustez do processo de treinamento, ampliando as dificuldades de convergência. Tal achado sustenta a necessidade de políticas de QoS que mitiguem oscilações e assegurem previsibilidade do fluxo de dados, sobretudo em aplicações de missão crítica.

No conjunto, os achados confirmam a hipótese formulada na Introdução (Capítulo 1): a PLR exerce impacto mais severo do que a latência isolada sobre a convergência do SL. Ao mesmo tempo, a dissertação aborda a lacuna destacada na revisão de literatura (Capítulo 2) ao analisar, de modo integrado, métricas de rede realistas e desempenho de aprendizado distribuído em cenários B5G/6G. A principal contribuição reside em oferecer uma instrumentação replicável que alinha métricas de comunicação e desempenho de SL, apoiando decisões de *codesign* entre algoritmos de aprendizado e políticas de alocação de recursos.

Como desdobramento natural, o *framework* posiciona-se para integração com plataformas abertas como a *Open Radio Access Network* (O-RAN), viabilizando conexão com *RAN Intelligent Controller*s (RICs) e a implementação de *xApps*/*rApps* orientadas a aprendizado de máquina, inclusive em ambientes programáveis como o ns-O-RAN [Lacava et al. 2023]. Tal direção aproxima a pesquisa acadêmica de *testbeds* reprodutíveis e amplia o potencial de validação em setores críticos (saúde, veículos autônomos e Indústria 4.0).

Por fim, reconhece-se que desafios estruturais do 5G permanecem: confiabilidade na ordem de 99,9999%, manutenção de latência ultrabaixa em escala e integração massiva de dispositivos heterogêneos [Kurose e Ross 2020]. A arquitetura aqui proposta representa um passo inicial em direção ao B5G e ao 6G: uma base sólida, científica e reprodutível, sobre a qual novas políticas de QoS, esquemas de fatiamento e estratégias de aprendizado colaborativo poderão ser projetadas e avaliadas com rigor.

5.3 Trabalhos Futuros

A evolução desta pesquisa pode seguir diferentes caminhos:

- Comparação entre SL e FL – investigar cenários com alta variabilidade de rede, avaliando se a maior robustez do FL compensa sua carga computacional;
- Uso de datasets complexos – adotar bases como CIFAR-10 e *ImageNet*, que ampliam os desafios de volume e variabilidade de classes, aproximando os testes de aplicações multimídia reais;
- Cenários multi-gNB com mobilidade e *handover* – explorar a escalabilidade do *framework* em topologias amplas, incorporando modelos realistas de mobilidade e avaliando os efeitos de *handover* em métricas como latência e *jitter*;
- Impacto da variabilidade de rede – analisar como oscilações em atraso, perdas de pacotes e disponibilidade de banda afetam a acurácia e o tempo de convergência, propondo mecanismos de resiliência;
- Integração com plataformas O-RAN – conectar o *framework* a (RICs), permitindo implementação de *xApps* e *rApps* para alocação dinâmica de recursos orientada a aprendizado de máquina;
- Políticas de alocação e eficiência energética – avaliar algoritmos de escalonamento capazes de equilibrar desempenho e consumo, aspecto crítico para dispositivos IoT e aplicações industriais;
- Segurança e privacidade em SL – investigar riscos de vazamento de dados na divisão do modelo, propondo soluções de criptografia e anonimização;
- Extensão para cenários 6G – adaptar a arquitetura a novos requisitos, como sensoriamento integrado, comunicação holográfica e uso de frequências em *terahertz*;
- Testes em *hardware* real – validar o *framework* em *testbeds* físicos ou dispositivos embarcados, aproximando os resultados de condições reais de operação.

É importante destacar que, nesta dissertação, optou-se pelo uso do *dataset* MNIST em vez de conjuntos mais complexos como o CIFAR-10. A escolha foi guiada pela necessidade de manter o foco na análise das métricas de rede e na validação da arquitetura proposta, assegurando simplicidade experimental e tempos de execução viáveis no ambiente computacional disponível. O MNIST, por conter apenas 10 classes de dígitos manuscritos e baixa dimensionalidade, viabilizou execuções rápidas e reprodutíveis, permitindo isolar com clareza os efeitos da latência, da PLR, da vazão e do consumo energético sobre o processo de aprendizado distribuído. Assim, a adoção do CIFAR-10 é projetada como trabalho futuro justamente por trazer maior variabilidade de classes, volume de dados e carga de processamento, ampliando os desafios e aproximando os experimentos de aplicações multimídia mais realistas.

Referências Bibliográficas

[3GPP 2019]3GPP. *3rd generation partnership project; technical specification group services and system aspects; system architecture for the 5G system stage 2;(Release 16): TS23. 501 V16. 3.0*. 2019.

[3GPP TR 38.802 2017]3GPP TR 38.802. *Study on New Radio (NR); Physical Layer Aspects (TR 38.802)*. 2017. https://www.3gpp.org/ftp/Specs/archive/38_series/38.802/. Version 14.2.0, Sep. 2017.

[3GPP TS 23.501 2025]3GPP TS 23.501. 3GPP TS, *System architecture for the 5G System (5GS)*. 2025. https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/18.10.00_60/ts_123501v181000p.pdf. Release 18. Accessed 2025-10-07.

[3GPP TS 23.502 2025]3GPP TS 23.502. 3GPP TS, *Procedures for the 5G System (5GS)*. 2025. https://www.etsi.org/deliver/etsi_ts/123500_123599/123502/18.05.00_60/ts_123502v180500p.pdf. Release 18. Accessed 2025-10-07.

[3GPP TS 38.211 2024]3GPP TS 38.211. 3GPP TS, *NR; Physical channels and modulation*. 2024. https://www.etsi.org/deliver/etsi_ts/138200_138299/138211/18.02.00_60/ts_138211v180200p.pdf. Release 18. Accessed 2025-10-07.

[3GPP TS 38.213 2022]3GPP TS 38.213. 3GPP TS, *NR; Physical layer procedures for control*. 2022. https://www.etsi.org/deliver/etsi_ts/138200_138299/138213/17.01.00_60/ts_138213v170100p.pdf. Release 17. Accessed 2025-10-07.

[3GPP TS 38.214 2025]3GPP TS 38.214. 3GPP TS, *NR; Physical layer procedures for data*. 2025. https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/18.06.00_60/ts_138214v180600p.pdf. Release 18. Accessed 2025-10-07.

[3GPP TS 38.300 2022]3GPP TS 38.300. 3GPP TS, *NR; NR and NG-RAN Overall description; Stage-2*. 2022. https://www.etsi.org/deliver/etsi_ts/138300_138399/138300/17.00.00_60/ts_138300v170000p.pdf. Release 17. Accessed 2025-10-07.

[3GPP TS 38.912 2018]3GPP TS 38.912. Technical Report, *Study on energy efficiency aspects of 5G*. 2018. https://www.etsi.org/deliver/etsi_tr/138900_138999/

138912/15.00.00_60/tr_138912v150000p.pdf. V15.0.0 (Release 15); latest version available on ETSI Deliver.

[Alfadhli et al. 2019]ALFADHLI, Y. et al. *Latency performance analysis of low layers function split for URLLC applications in 5G networks*. 2019. 106865 p. https://linkinghub.elsevier.com/retrieve/pii/S1389128619301343.

[Ali et al. 2021]ALI, Z. et al. *3GPP NR V2X Mode 2: Overview, Models and System-Level Evaluation*. 2021. 89554-89579 p. https://doi.org/10.1109/ACCESS.2021.3090855.

[Amjad et al. 2021]AMJAD, Z. et al. *Latency reduction for narrowband URLLC networks: a performance evaluation*. 2021. 2577–2593 p. https://link.springer.com/10.1007/s11276-021-02553-x.

[Ayad, Renner e Schmeink 2021]AYAD, A.; RENNER, M.; SCHMEINK, A. *Improving the Communication and Computation Efficiency of Split Learning for IoT Applications*. [S.l.]: IEEE, 2021. 01–06 p. https://ieeexplore.ieee.org/document/9685493/.

[Campanile et al. 2020]CAMPANILE, L. et al. *Computer network simulation with ns-3: A systematic literature review*. [S.l.]: MDPI, 2020. 272 p.

[Chen, Li e Chakrabarti 2021]CHEN, X.; LI, J.; CHAKRABARTI, C. *Communication and Computation Reduction for Split Learning using Asynchronous Training*. [S.l.]: IEEE, 2021. 76–81 p. https://ieeexplore.ieee.org/document/9605049/.

[CTTC 2023]CTTC. *NR Module - 5G LENA*. 2023. https://5g-lena.cttc.es/static/archive/NR_V2X_V0.1_doc.pdf. Acesso em: 04 ago. 2025.

[CTTC 2025]CTTC. *Features | 5G LENA*. 2025. https://5g-lena.cttc.es/features/. Acesso em: 04 ago. 2025.

[Dachille, Huang e Liu 2024]DACHILLE, J.; HUANG, Y.; LIU, Z. *The Impact of Cut Layer Selection in Split Federated Learning*. 2024. https://arxiv.org/abs/2412.15536.

[Duan et al. 2022]DUAN, Q. et al. *Combined Federated and Split Learning in Edge Computing for Ubiquitous Intelligence in Internet of Things: State-of-the-Art and Future Directions*. 2022. 5983 p. https://www.mdpi.com/1424-8220/22/16/5983.

[Ebrahimi et al. 2024]EBRAHIMI, S. et al. From Domain 5G to End-to-End 6G Network Slicing: A Survey. *IEEE Communications Surveys & Tutorials*, 2024. Open version. Accessed 2025-10-07.

[Ekaireb et al. 2022]EKAIREB, E. et al. *ns3-fl: Simulating Federated Learning with ns-3*. 2022. 97–104 p.

[Itahara, Nishio e Yamamoto 2021]ITAHARA, S.; NISHIO, T.; YAMAMOTO, K. *Packet-Loss-Tolerant Split Inference for Delay-Sensitive Deep Learning in Lossy Wireless Networks*. [S.l.]: arXiv, 2021. http://arxiv.org/abs/2104.13629.

[Khan et al. 2022]KHAN, B. S. et al. *URLLC and eMBB in 5G Industrial IoT: A Survey*. 2022. 1134–1163 p. https://ieeexplore.ieee.org/document/9826826/.

[Koda et al. 2020]KODA, Y. et al. *Communication-Efficient Multimodal Split Learning for mmWave Received Power Prediction*. 2020. 1284–1288 p. https://ieeexplore.ieee.org/document/9026781/.

[Koutlia e al. 2024]KOUTLIA, K.; AL. et. *5G NR Tutorial*. 2024. Tutorial, 2024. Disponível em apresentação técnica sobre 5G NR.

[Koutlia et al. 2023]KOUTLIA, K. et al. *System analysis of QoS schedulers for XR traffic in 5G NR*. 2023. 102745 p. https://linkinghub.elsevier.com/retrieve/pii/S1569190X23000230.

[Kurose 2021]KUROSE, J. F. *Redes de computadores e a internet: uma abordagem top-down*. [S.l.]: Bookman, 2021.

[Kurose e Ross 2020]KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. 8. ed. Boston: Pearson, 2020. Disponível em: https://www.pearson.com. Acesso em: 18 set. 2025. ISBN 9780135927382.

[LABORA-INF/UFG 2025]LABORA-INF/UFG. *SplitLearning-B5G*. [S.l.]: GitHub repository, 2025. https://github.com/LABORA-INF-UFG/SplitLearning-B5G.

[Lacava et al. 2023]LACAVA, A. et al. *ns-O-RAN: Simulating O-RAN 5G Systems in ns-3*. 2023. 1–10 p.

[Lagén et al. 2023]LAGéN, S. et al. *QoS Management for XR Traffic in 5G NR: A Multi-Layer System View and End-to-End Evaluation*. 2023. 192-198 p. https://doi.org/10.1109/MCOM.015.2200745.

[Larrañaga et al. 2023]LARRAñAGA, A. et al. *An open-source implementation and validation of 5G NR configured grant for URLLC in ns-3 5G LENA: A scheduling case study in industry 4.0 scenarios*. 2023. 103638 p. https://www.sciencedirect.com/science/article/pii/S1084804523000577.

[Lin et al. 2024]LIN, Z. et al. *Split Learning in 6G Edge Networks*. [S.l.]: arXiv, 2024. http://arxiv.org/abs/2306.12194.

[Liu et al. 2023]LIU, P. et al. *Towards 5G new radio sidelink communications: A versatile link-level simulator and performance evaluation*. 2023. 231-243 p. https://doi.org/10.1016/j.comcom.2023.06.005.

[Liu, Deng e Mahmoodi 2023]LIU, X.; DENG, Y.; MAHMOODI, T. *Wireless Distributed Learning: A New Hybrid Split and Federated Learning Approach*. 2023. 2650–2665 p. https://ieeexplore.ieee.org/document/9923620/.

[Luna, Ferreira e Rocha 2021]LUNA, C. A.; FERREIRA, J. P.; ROCHA, D. M. *AI-based scheduling optimization for 5G networks*. 2021. 7453–7467 p.

[Luna, Oliveira e Silva 2021]LUNA, F.; OLIVEIRA, J.; SILVA, R. *Implementação de network slicing no 5G NR com perfis eMBB, URLLC e mMTC*. Fortaleza, CE: SBrT, 2021. 1–5 p. https://sbrt.org.br.

[Matsubara, Levorato e Restuccia 2022]MATSUBARA, Y.; LEVORATO, M.; RESTUCCIA, F. *Split Computing and Early Exiting for Deep Learning Applications: Survey and R journal = ACM Computing Surveys*. 2022. 1–30 p. https://dl.acm.org/doi/10.1145/3527155.

[Medeiros, Santos e Almeida 2022]MEDEIROS, P.; SANTOS, T.; ALMEIDA, L. *Scheduler customizado para priorização de tráfego URLLC em cenários de slicing 5G*. Campinas, SP: SBrT, 2022. 1–6 p. https://sbrt.org.br.

[Morocho-Cayamcela, Lee e Lim 2019]MOROCHO-CAYAMCELA, M. E.; LEE, H.; LIM, W. *Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions*. 2019. 137184–137206 p. https://ieeexplore.ieee.org/document/8844682/.

[Nakashima et al. 2022]NAKASHIMA, T. et al. *ns3-ai: Rate control for wireless lan by deep q-network*. [S.l.]: The Institute of Electronics, Information and Communication Engineers, 2022.

[NGMN Alliance 2015]NGMN Alliance. *NGMN 5G White Paper*. 2015. https://www.ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf. Version 1.0. Accessed 2025-10-07.

[ns-3 Project 2025]ns-3 Project. *Organization — ns-3 Manual*. 2025. https://www.nsnam.org/docs/manual/html/organization.html. Acesso em: 20 set. 2025.

[Oh et al. 2022]OH, S. et al. *LocFedMix-SL: Localize, Federate, and Mix for Improved Scalability, Convergence, and Latency in Split Learning*. [S.l.]: ACM, 2022. 3347–3357 p. https://dl.acm.org/doi/10.1145/3485447.3512153.

[Park et al. 2022]PARK, J. et al. *Extreme ultra-reliable and low-latency communication*. 2022. 133–141 p. https://www.nature.com/articles/s41928-022-00728-8.

[Popovski et al. 2018]POPOVSKI, P. et al. *5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View*. 2018. 55765–55779 p. https://ieeexplore.ieee.org/document/8476595/.

[Ryu, Won e Lee 2022]RYU, J.; WON, D.; LEE, Y. *A Study of Split Learning Model*. 2022. 1–4 p.

[Shiranthika et al. 2023]SHIRANTHIKA, C. et al. *SplitFed resilience to packet loss: Where to split, that is the question*. [S.l.]: arXiv, 2023. http://arxiv.org/abs/2307.13851.

[Shiranthika, Saeedi e Bajić 2023]SHIRANTHIKA, C.; SAEEDI, P.; BAJIć, I. V. *Decentralized Learning in Healthcare: A Review of Emerging Techniques*. 2023. 54188–54209 p. https://ieeexplore.ieee.org/document/10141615/.

[Shruthi 2024]SHRUTHI, K. *A Deep Learning Approach to Find Optimal Path in Underwater Networks Using ns3-ai*. 2024. https://doi.org/10.21203/rs.3.rs-4235108/v1.

[Sousa 2022]SOUSA, L. S. d. *Dissertation in 5G Network Slicing*. Goiânia, Brasil: [s.n.], 2022. file-KCgzxYC7KYHyeaqYYybLNW. Dissertação de Mestrado.

[Thapa et al. 2022]THAPA, C. et al. *SplitFed: When Federated Learning Meets Split Learning*. 2022. 8485–8493 p. https://ojs.aaai.org/index.php/AAAI/article/view/20825.

[Vepakomma et al. 2018]VEPAKOMMA, P. et al. *Split learning for health: Distributed deep learning without sharing raw patient data*. [S.l.]: arXiv, 2018. http://arxiv.org/abs/1812.00564.

[Villegas e Costa 2024]VILLEGAS, M. R.; COSTA, F. J. *Low-latency aware scheduling algorithms for URLLC in 5G systems*. 2024. 1–7 p.

[Wang et al. 2023]WANG, Z. et al. *Privacy-Preserving Split Learning for Large-Scaled Vision Pre-Training*. 2023. 1539–1553 p. https://ieeexplore.ieee.org/document/10041745/.

[Wu et al. 2023]WU, W. et al. Split learning over wireless networks: Parallel design and resource management. v. 41, n. 4, p. 1051–1066, 2023. ISSN 0733-8716, 1558-0008.

[Xu et al. 2024]XU, C. et al. *Accelerating Split Federated Learning Over Wireless Communication Networks*. 2024. 5587–5599 p. https://ieeexplore.ieee.org/document/10304624/.

[Yin et al. 2020]YIN, H. et al. *ns3-ai: Fostering artificial intelligence algorithms for networking research*. 2020. 57–64 p. URL:https://dl.acm.org/doi/abs/10.1145/3389400.3389404.

[Yu et al. 2024]YU, H. et al. *Distributed Split Learning for Map-Based Signal Strength Prediction Empowered by Deep Vision Transformer*. 2024. 2358–2373 p. https://ieeexplore.ieee.org/document/10266792/.

[Zhang et al. 2023]ZHANG, Z. et al. *Privacy and Efficiency of Communications in Federated Split Learning*. 2023. 1380–1391 p. https://ieeexplore.ieee.org/document/10138067/.

APÊNDICE A

Configuração do Ambiente de Simulação

Nas simulações e treinos, foi utilizado equipamento com as seguintes configurações:

- **Hardware:** notebook com processador Intel Core i5-2410M CPU @ 2.30 GHz, 8 GB de RAM, sem GPU dedicada.
- **Sistema Operacional:** Linux Mint 21.3 Cinnamon, Kernel 5.15.0-116-generic.

Configurações do Software

Tabela A.1: Configuração do ambiente de execução

Componente	**Versão / Configuração**
ns-3	3.45 (release oficial)
5G-LENA	CTTC NR-v4.1
ns3-ai	v1.2 (integrado ao ns-3.45)
Compilador	GCC 10.5.0
CMake	4.1.1
Python	3.10.6
PyTorch	2.7.0
NumPy	2.2.6
Pandas	2.2.3
Matplotlib	3.10.5
SciPy	1.15.3

Foram listados apenas os pacotes mais relevantes para a reprodução dos experimentos. A listagem completa das dependências *Python* está disponível no repositório oficial do projeto [LABORA-INF/UFG 2025].