



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Solução para Monitoramento Adaptativo de Métricas de Redes *Open* RAN

Matheus Fagundes de Souza Dória

Orientador: Prof. Dr. Vicente A. de Sousa Jr.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Telecomunicações) como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica e de Computação.

Número de ordem PPgEEC: M694
Natal, RN, Agosto de 2025

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Dória, Matheus Fagundes de Souza.

Solução para monitoramento adaptativo de métricas de Redes
Open RAN / Matheus Fagundes de Souza Doria. - 2025.
52 f.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande
do Norte, Centro de Tecnologia, Programa de Pós-graduação em
Engenharia Elétrica e de Computação, Natal, 2025.

Orientação: Prof. Dr. Vicente Angelo de Sousa Junior.

Coorientação: Prof. Dr. Allan de Medeiros Martins.

1. 5G - Dissertação. 2. Open RAN - Dissertação. 3. xApp -
Dissertação. 4. Monitoramento de KPIs - Dissertação. 5. Economia
de energia - Dissertação. I. Sousa Junior, Vicente Angelo de.
II. Martins, Allan de Medeiros. III. Título.

RN/UF/BCZM

CDU 621:004

Elaborado por Jackeline dos Santos Pinheiro da Silva Maia
Cavalcanti - CRB-15/317

Agradecimentos

A Deus, por me iluminar, guiar meus passos, fortalecer-me nos momentos de dificuldade e permitir que eu alcance os meus objetivos.

Ao meu orientador e amigo, professor Vicente Angelo, pelo apoio e por todos os ensinamentos transmitidos ao longo desta jornada. Também por ter contribuído de forma significativa para o meu crescimento, tanto profissional quanto pessoal.

À minha família, especialmente aos meus pais, Hermann e Christianne, por todo apoio, carinho, amparo e compreensão. Agradeço também por serem exemplos de dedicação e caráter, que levo comigo como inspiração.

À minha namorada, Laura Varela, por todo apoio, incentivo e compreensão nos momentos de ausência. Agradeço também, por estar sempre ao meu lado nessa caminhada, fazendo-se presente mesmo quando a distância era inevitável.

Ao meu amigo Carlos Antonio, pelas madrugadas de estudo, pelos trabalhos em conjunto e pela parceria constante ao longo de toda a jornada da graduação, do intercâmbio e do mestrado.

A todos os amigos do LANCE, GppCom e Regina, por me proporcionarem o privilégio de aprender e evoluir diariamente ao lado de vocês. Agradeço também por toda a contribuição, parceria e pelos momentos de descontração ao longo dessa trajetória.

Resumo

O monitoramento em redes Open RAN prevê uma troca contínua de métricas entre *xApps* e a rede de acesso de rádio por meio da interface E2. Devido à intensa troca de sinalização, esse monitoramento consome uma quantidade significativa de recursos (de *hardware* e de rede) e pode impactar negativamente a eficiência do sistema. Este trabalho propõe o desenvolvimento de um *xApp* para monitoramento adaptativo de métricas, ajustando a frequência de coleta de *Key Performance Indicators* (KPIs) com base em análise em tempo real. A proposta é baseada na definição de uma Análise de Risco e busca reduzir o consumo excessivo de recursos sem comprometer a qualidade do monitoramento, contribuindo para uma gestão mais eficiente e econômica de redes *Open RAN*. O *xApp* é implementado utilizando uma infraestrutura baseada em *software* de código aberto, sendo o FlexRIC a plataforma utilizada no seu desenvolvimento. Para a implementação do núcleo de rede e da RAN 5G, utilizou-se o OpenAirInterface, integrado a uma plataforma de rádio definido por *software* (SDR), viabilizando o uso da RAN com um equipamento do usuário do tipo COTS (*Commercial-off-the-shelf*). Os resultados são promissores ao indicarem que o monitoramento adaptativo de métricas oferece possibilidade de economia de energia comparado ao monitoramento com frequência fixa. Além disso, o ganho energético proporcionado pelo monitoramento proposto torna-se ainda mais relevante quando inserido em um contexto de larga escala, no qual a sobrecarga gerada pela coleta de métricas via interface E2 representa um desafio significativo para a eficiência do sistema.

Palavras-chave: 5G; *Open RAN*; *xApp*; Monitoramento de KPIs; Economia de energia.

Abstract

Monitoring in Open RAN networks involves a continuous exchange of metrics between xApps and the radio access network through the E2 interface. Due to the intense signaling exchange, this monitoring consumes a significant amount of resources (both hardware and network) and can negatively impact system efficiency. This work proposes the development of an xApp for adaptive metric monitoring, adjusting the sampling frequency of Key Performance Indicators (KPIs) based on real-time analysis. The proposal is based on the definition of a Risk Analysis and aims to reduce excessive resource consumption without compromising monitoring quality, contributing to more efficient and cost-effective Open RAN network management. The xApp is implemented using an open-source software-based infrastructure, with FlexRIC as the development platform. For the implementation of the 5G core network and RAN, OpenAirInterface was used, integrated with a Software Defined Radio (SDR) platform, enabling the use of the RAN with Commercial-off-the-shelf (COTS) user equipment. The results are promising, indicating that adaptive metric monitoring offers potential energy savings compared to fixed-frequency monitoring. Furthermore, the energy gain provided by the proposed monitoring becomes even more relevant when applied to a large-scale context, in which the overhead generated by metric collection via the E2 interface poses a significant challenge to system efficiency.

Keywords: 5G; *Open RAN*; xApp; KPIs monitoring; Energy saving.

Sumário

Sumário	i
Lista de Figuras	ii
Lista de Tabelas	iii
Lista de Siglas	iv
1 Introdução	1
1.1 Escopo de Realização, Hipótese Científica e Objetivos do Trabalho	3
1.2 Trabalhos Relacionados	4
1.2.1 Investigação Quantitativa	4
1.2.2 Investigação Qualitativa	6
1.3 Organização da Dissertação	8
2 Redes <i>Open</i> RAN	9
2.1 Arquitetura de Referência da O-RAN <i>Alliance</i>	9
2.2 Sobrecarga de Monitoramento via Interface E2	13
3 Solução Proposta: <i>xApp</i> adaptativo	15
3.1 Implementação do <i>xApp</i> adaptativo	18
4 Metodologia de Prova de Conceito	21
4.1 <i>Setup</i> e Cenário de Medição	21
4.2 Procedimentos de Medição	25
5 Resultados	30
5.1 Definição do Limiar de Decisão	30
5.2 Avaliação de Desempenho	34
6 Conclusão	38
6.1 Produção Acadêmica	39
Referências bibliográficas	40

Lista de Figuras

1.1	Projeção do mercado global de <i>Open RAN</i> [1].	2
1.2	Publicações por ano para o Grupo #0.	5
1.3	Publicações por ano para o Grupo #1.	5
2.1	Arquitetura 5G RAN com split CU/DU.	10
2.2	RAN proprietária.	10
2.3	RAN aberta.	11
2.4	Arquitetura lógica simplificada da O-RAN [2].	12
3.1	Diagrama de bloco da fase <i>KPI Evaluation Phase</i> (KEP).	16
3.2	Diagrama de bloco da fase <i>Sampling Time Definition Phase</i> (SDP).	16
3.3	Diagrama de bloco do xApp Adaptativo.	17
3.4	Composição de <i>KPI Evaluation Phase</i> (KEP) e <i>Sampling Time Definition Phase</i> (SDP) no <i>Duty-cycle</i>	18
3.5	Diagrama de fluxo do algoritmo.	20
4.1	Setup de medição.	22
4.2	Fluxo de mensagens para coleta de métricas via xApp no FlexRIC.	23
4.3	Sistema de monitoramento via Zabbix.	24
4.4	Cenário de medição.	25
4.5	Diagrama da definição do limiar de decisão.	26
4.6	Esboço do grid de medição para diferentes valores de DC e α	27
4.7	Diagrama para definição de acurácia.	29
5.1	Distribuição estatística dos valores de σ_{Δ} para definição de θ	31
5.2	Funcionamento adaptativo: σ_{Δ} e troca de frequência de amostragem.	31
5.3	<i>Signal to Noise Ratio</i> (SNR) do canal <i>Physical Uplink Shared Channel</i> (PUSCH) em função do tempo.	32
5.4	xApp adaptativo operando com alta frequência de amostragem.	33
5.5	xApp adaptativo operando com frequência de amostragem reduzida.	34
5.6	Grid de economia de CPU - xApp Adaptativo vs Estático de 1ms.	35

Lista de Tabelas

1.1	Número de artigos por grupo de palavras-chave.	4
4.1	Mapeamento entre intervalos de SNR e índices de Esquema de Modulação e Codificação (MCS).	28
5.1	Compromisso entre economia de CPU e acurácia na decisão de MCS. . .	36

Lista de Siglas

- Near-RT RIC** *Near-Real-Time RIC*. 3, 11–14, 21
Non-RT RIC *Non-Real-Time RIC*. 3, 11
- CDF** *Função de Distribuição Acumulada*. 26, 30, 31
COTS *Commercial Off-The-Shelf*. 22
CU *Central Unit*. 9–11
- DU** *Distributed Unit*. 9–11
- eMBB** *enhanced Mobile Broadband*. 1
- gNB** *Next Generation Node B*. 14
- IA** *Inteligência Artificial*. 6
IoT *Internet of Things*. 1
ITU *International Telecommunication Union*. 1
- KEP** *KPI Evaluation Phase*. ii, 15–19, 25, 27, 34
KPI *Key Performance Indicator*. 3, 6, 7, 12–16, 22–24, 27, 31, 33, 35–38
- MAC** *Medium Access Control*. 9, 13, 19, 27
MCS *Esquema de Modulação e Codificação*. iii, 28, 35, 36
MIMO *Multiple Input Multiple Output*. 1
mMTC *massive Machine-Type Communications*. 1
- NAS** *Non-Access-Stratum*. 9
- OAI** *OpenAirInterface*. 18, 19, 21–23
- PDCP** *Packet Data Convergence Protocol*. 9, 13, 19
PDU *Packet Data Unit*. 9
PHR *Power Headroom Report*. 13, 19
PHY *Physical*. 9, 13
PUSCH *Physical Uplink Shared Channel*. ii, 28, 30–32, 35
- QoS** *Quality of Service*. 1
- RAN** *Rede de Acesso de Rádio*. 1–6, 9–13, 18, 21, 23, 38
RIC *RAN Intelligent Controller*. 2, 3, 6, 13, 14, 18, 21
RLC *Radio Link Control*. 9, 13, 19

RRC *Radio Resource Control*. 9, 14
RRM Gerência de Recursos de Rádio. 2, 3
RU *Radio Unit*. 9–11, 21

SDAP *Service Data Adaption Protocol*. 9
SDP *Sampling Time Definition Phase*. ii, 16–20, 27
SDR Rádio definido por *Software*. 21, 22
SMO *Service Management and Orchestration*. 11, 12
SNR *Signal to Noise Ratio*. ii, iii, 28, 30–33, 35

UE Equipamento de Usuário. 3, 6, 19, 21, 25, 37
URLLC *Ultra-Reliable Low-Latency Communications*. 1
USIM *Universal Subscriber Identity Module*. 22

Capítulo 1

Introdução

O avanço das redes móveis tem sido marcado pela crescente complexidade de suas arquiteturas e pela diversidade de aplicações que exigem conectividade confiável, de baixa latência e altamente escalável. A quinta geração de redes móveis (5G) se destaca por introduzir novas possibilidades técnicas e comerciais, impulsionadas por requisitos como fatiamento de rede, *Multiple Input Multiple Output* (MIMO) massivo, suporte a múltiplas bandas e tecnologias, bem como o uso intensivo de inteligência artificial na operação da rede. Entretanto, ao mesmo tempo em que essas tecnologias aumentam a capacidade e flexibilidade do sistema, elas também elevam a complexidade de gerenciamento e os custos operacionais das operadoras [3, 4].

Enquanto as gerações passadas foram projetadas principalmente para conectar pessoas por meio de *smartphones* e *tablets*, o 5G foi concebido para conectar tudo, abrangendo uma gama muito maior de dispositivos e aplicações. Com sua arquitetura de rede flexível, o 5G introduz características inovadoras, como velocidades ultrarrápidas, baixa latência e alta confiabilidade, sendo capaz de atender a diversos casos de uso, os quais foram definidos pela *International Telecommunication Union* (ITU) [5], como: *enhanced Mobile Broadband* (eMBB), *massive Machine-Type Communications* (mMTC) e *Ultra-Reliable Low-Latency Communications* (URLLC), cada um com demandas específicas de *Quality of Service* (QoS). O eMBB busca oferecer um aumento expressivo na taxa de transmissão de dados ao usuário final; já o mMTC foca na conexão simultânea de um grande número de dispositivos; enquanto o URLLC visa garantir comunicações com altíssima confiabilidade e latência mínima [6, 7].

Os aspectos desses três casos de uso possibilitam o suporte a uma ampla variedade de aplicações, abrangendo desde cenários imersivos de realidade virtual até redes de dispositivos de *Internet of Things* (IoT). Contudo, apesar de oferecer grande potencial para o desenvolvimento de novas soluções e serviços inovadores, essa evolução acarreta no alto custo de implantação e operação da rede, na densificação da mesma e no surgimento de aplicações robustas cada vez mais exigentes [8, 9]. Esses aspectos surgem como possíveis barreiras para a adoção massiva de soluções baseadas no 5G, principalmente em ambientes com alta exigência de QoS.

Diante desse cenário desafiador, ganha destaque a atuação da *O-RAN Alliance* [10], uma organização internacional que propõe um novo paradigma para a Rede de Acesso de Rádio (RAN). Fundada com o objetivo de acelerar a inovação, a O-RAN propõe uma

arquitetura aberta, desagregada e interoperável, promovendo a separação entre *hardware* e *software*, e incentivando a participação de múltiplos fornecedores.

Esse paradigma vem sendo adotado mundialmente e o mercado global de RAN aberta está em rápido crescimento. Como apresentado na Figura 1.1, estima-se que o tamanho desse mercado deve atingir 13,39 bilhões de dólares até 2030, chegando a aproximadamente 38,71 bilhões até 2034 [1]. No Brasil, o programa OpenRAN@Brasil [11], conduzido pela RNP e pelo CPqD, apoia-se em recursos federais e investimentos de aproximadamente 180 milhões de reais, voltados para acelerar a pesquisa e o desenvolvimento em 5G, 6G e *Open RAN* [12]. Atualmente, o programa está iniciando sua Fase 3, a qual tem o objetivo de expandir a infraestrutura nacional de experimentação em redes *Open RAN* para as regiões Norte, Nordeste e Sul [11].

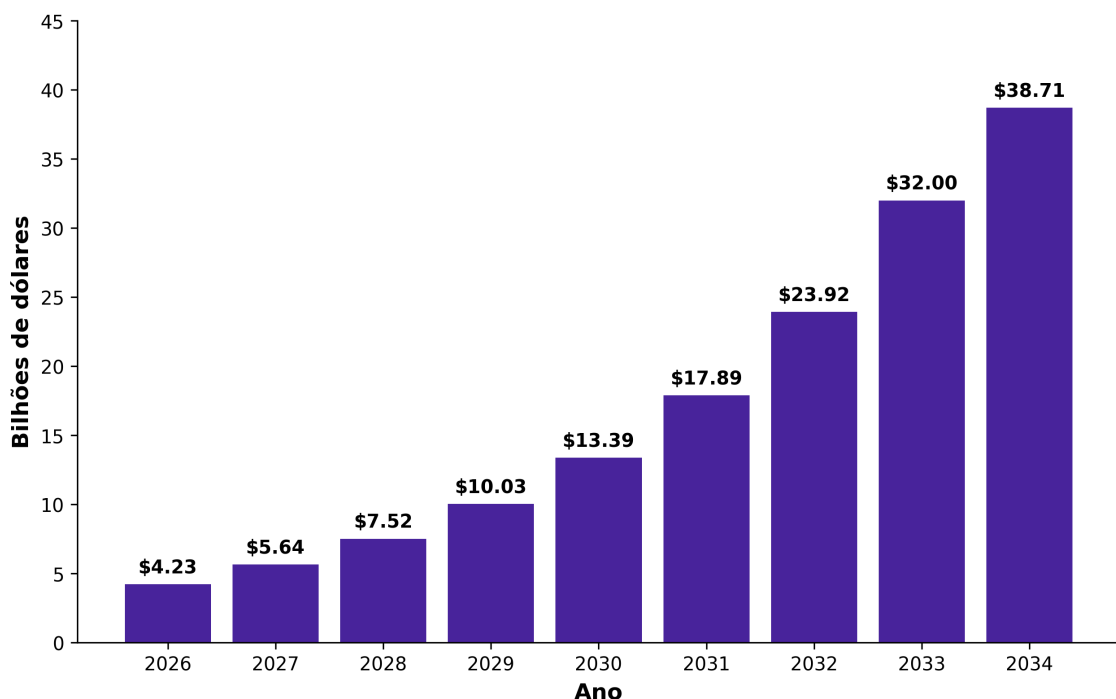


Figura 1.1: Projeção do mercado global de *Open RAN* [1].

Entre os principais pilares das redes *Open RAN*, destacam-se duas premissas principais:

1. Promover uma RAN com interfaces padronizadas e abertas, promovendo flexibilidade e interoperabilidade entre equipamentos de diferentes fornecedores. Além de baratear seu custo de implantação;
2. Incorporar inteligência à rede por meio de controladores especializados, visando otimizar a Gerência de Recursos de Rádio (RRM), melhorar a eficiência energética e possibilitar uma gestão adaptativa da rede [8].

Como ator central nessa proposta, está o controlador inteligente da RAN, conhecido como *RAN Intelligent Controller* (RIC). Ele fornece mecanismos de controle em duas escalas de tempo, por meio de dois componentes:

1. *Non-Real-Time* RIC (*Non-RT* RIC), responsável por um loop de controle de funções com latência superior a um segundo; Neste componente operam os *rApps*, que são aplicativos responsáveis por desempenhar funções de aprendizado de máquina, decisões estratégicas e políticas de longo prazo [8, 13];
2. *Near-Real-Time* RIC (*Near-RT* RIC), responsável por um loop de controle de funções com latência que varia de dez milissegundos a um segundo. Associados ao *Near-RT* RIC estão os *xApps*, que são aplicativos que desempenham ações específicas para auxiliar no gerenciamento de recursos de rádio, como controle de interferência, balanceamento de carga e *handover*. [8, 13].

A infraestrutura promovida pelo RIC permite que desenvolvedores externos implementem seus próprios algoritmos de RRM na forma de *xApps*, organizados como microsserviços capazes de monitorar métricas em tempo quase real e aplicar comandos diretamente sobre a rede. No entanto, apesar de seu potencial, o processo de monitoramento de métricas representa uma fonte significativa de consumo de CPU e energia, especialmente em cenários de larga escala [14].

1.1 Escopo de Realização, Hipótese Científica e Objetivos do Trabalho

Considerando o problema supracitado, este trabalho de mestrado possui como hipótese científica a redução do consumo excessivo de recursos por meio de um monitoramento adaptativo de *Key Performance Indicators* (KPIs) em sistemas reais 5G compatíveis com a O-RAN. Sendo assim, o Objetivo Geral da pesquisa é desenvolver um *xApp* de monitoramento de métricas da RAN 5G, com frequência de monitoramento adaptativa, levando em consideração a análise de KPIs na escala de tempo do *Near-RT* RIC, tornando o monitoramento mais econômico em determinados momentos.

Com base nesta hipótese científica, o Objetivo Geral será explorado por intermédio dos seguintes objetivos específicos:

- Construção de um *testbed* para desenvolvimento e avaliação de *xApps*. O *testbed* deve ser composto por pelo menos um núcleo de rede 5G, uma RAN e um Equipamento de Usuário (UE), além de ser integrado com RIC compatível com o da O-RAN;
- Concepção de um *xApp* para monitoramento adaptativo de KPIs, levando em consideração o compromisso entre economia de recursos de *hardware* e qualidade da métrica monitorada;
- Implementação do *xApp* e sua integração no *testbed* construído;
- Implementação de um sistema de monitoramento, a ser integrado ao *testbed*, com o objetivo de monitorar o *hardware* envolvido na execução do *xApp* (e do RIC) e da métrica alvo do monitoramento adaptativo implementado no *xApp*;
- Definição do cenário de avaliação do *xApp* proposto e implementação de *scripts* para automatização de campanhas de medição;

- Realização de uma análise de prova de conceito, comparando a solução proposta e o monitoramento com frequência fixa.

1.2 Trabalhos Relacionados

Esta seção apresenta uma pesquisa bibliográfica dos trabalhos relacionados a redes *Open RAN* e utilização de *xApps* com foco na economia de energia. Inicialmente, a Subseção 1.2.1 traz uma análise bibliométrica, com o objetivo de quantificar os trabalhos relacionados. A Subseção 1.2.2 disserta sobre os principais trabalhos selecionados no estudo quantitativo.

1.2.1 Investigação Quantitativa

Realizou-se um levantamento nas bases de dados *Scopus* e *Web of Science*, utilizando três combinações de palavras-chave. Cada combinação foi denominada **Grupo**, e as palavras-chave foram pesquisadas no título, no resumo ou nas palavras-chave dos artigos. A Tabela 1.1 apresenta a quantidade de artigos encontrados para cada grupo de palavras-chave e base de dados.

Tabela 1.1: Número de artigos por grupo de palavras-chave.

Grupo	Palavras-chave	Scopus	Web of Science
#0	“Open RAN” OR “OpenRAN” OR “O-RAN”	1.220	799
#1	#0 AND <i>xApp</i>	135	79
#2	#1 AND <i>Energy</i>	16	7

O Grupo #0 engloba o tema geral relacionado a este trabalho, abrangendo artigos publicados desde 2013, ano do primeiro artigo publicado sobre *Open RAN*, até julho de 2025. Essa pesquisa resultou em um total de 1.220 artigos na base *Scopus* e 799 na base *Web of Science*. A Figura 1.2 apresenta o número de artigos por ano para essa busca na base *Scopus*, que retornou a maior quantidade de trabalhos. Observa-se que o tema geral em que este trabalho está inserido ganhou destaque entre 2019 e 2020, com uma crescente demanda até 2024 e, até julho de 2025, foram registrados 212 artigos.

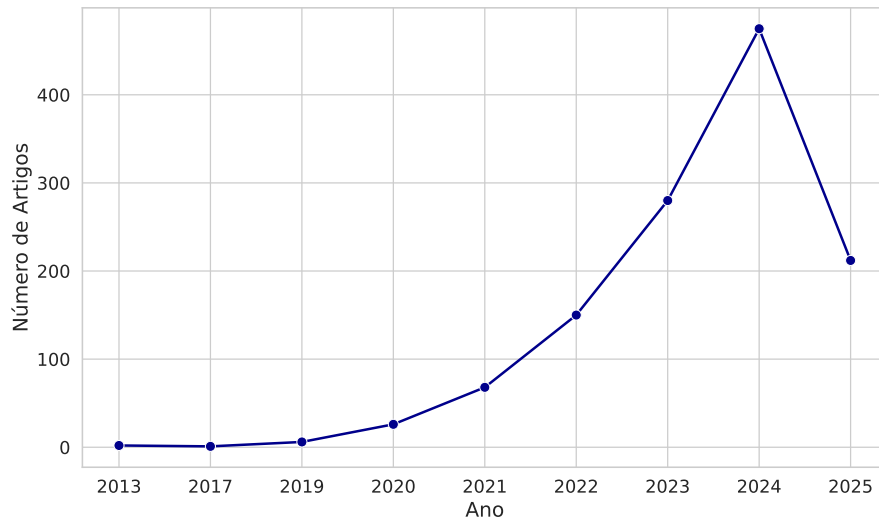


Figura 1.2: Publicações por ano para o Grupo #0.

Em seguida, o Grupo #1 foi criado com o objetivo de restringir o escopo de busca para trabalhos envolvendo o uso de *xApps*, acrescentando o termo *xApp*. Como resultado, a pesquisa trouxe um total de 135 artigos na base Scopus e 79 na *Web of Science*. Na Figura 1.3, resultado da busca na base Scopus para o Grupo #1, observa-se que os primeiros artigos sobre o tema começaram a ser publicados somente a partir de 2021, com um aumento significativo no número de publicações apenas em 2023. Isso pode ser explicado pelo fato de que, embora o tema *Open RAN* tenha se destacado entre 2019 e 2020, levou algum tempo até que as redes *Open RAN* estivessem suficientemente maduras para possibilitar o uso e o teste de *xApps*.

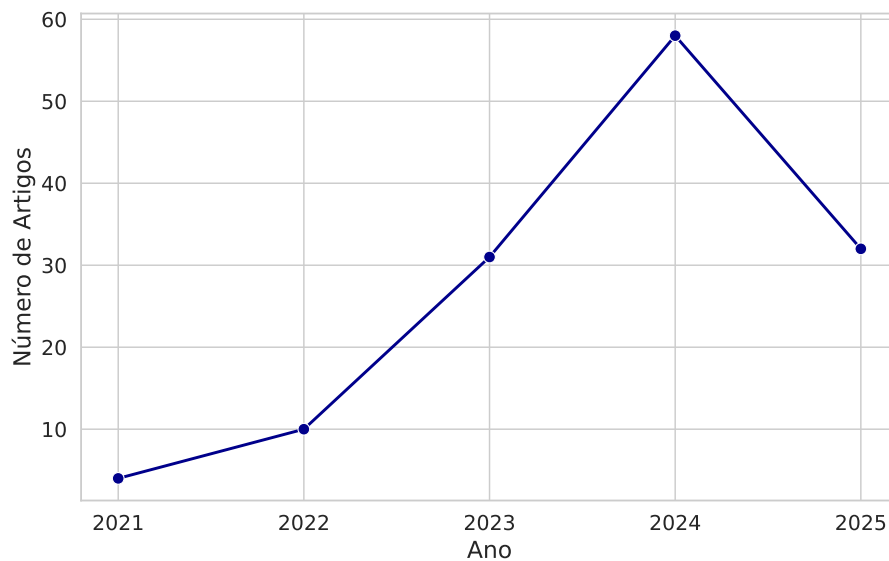


Figura 1.3: Publicações por ano para o Grupo #1.

Por fim, o Grupo #2 tem como objetivo restringir a busca para o tema relacionado diretamente com este trabalho, com foco na economia de energia. Esse grupo é composto pelos termos do Grupo #1, acrescidos da palavra *Energy*, que abrange diversas variações em inglês associadas a economia e ao consumo de energia. Nessa busca, os primeiros trabalhos foram publicados em 2023, totalizando apenas dois. Em 2024, o número de publicações triplica, alcançando seis artigos, e, até o primeiro semestre de 2025, já haviam sido publicados oito trabalhos. Esses números, quando comparados com os do Grupo #1, representam 6,45%, 10,35% e 25% do total de publicações dos anos de 2023, 2024 e 2025, respectivamente. Esse crescimento evidencia que a preocupação com o consumo energético no contexto de Open RAN é um tema recente, mas que vem ganhando relevância progressivamente.

1.2.2 Investigação Qualitativa

O levantamento dos principais trabalhos relacionados foi realizado a partir de uma seleção de trabalhos identificados pelo Grupo #1 na base de dados Scopus, que apresentou o maior número de artigos relevantes. Entre esses, foram selecionados os estudos que abordam a economia de recursos/energia baseada no uso de *xApps*.

O estudo desenvolvido em [14] evidencia, com medições reais, que o monitoramento de KPIs pode se tornar um gargalo no consumo energético dos RICs. Os resultados demonstram que tanto o volume de tráfego na interface E2 quanto a quantidade de KPIs reportados apresentam uma relação linear com o consumo de energia do RIC. Para mitigar esse problema, os autores propõem um mecanismo de otimização que evita transmissões redundantes de KPIs. Os resultados demonstram reduções significativas no consumo energético, chegando a 87% em alguns cenários, comprovando a necessidade de estratégias eficientes de monitoramento para garantir a escalabilidade e a sustentabilidade energética em arquiteturas *Open RAN*.

O trabalho [15] propõe dois *xApps* com foco na eficiência energética de redes O-RAN. Esses *xApps* ajustam o estado operacional dos rádios, alternando entre os modos *Active* e *Sleep*. O objetivo principal é configurar o rádio no modo *Sleep* sempre que possível, sem comprometer a taxa de dados exigida pelos UEs e garantindo o uso eficiente dos blocos de recursos disponíveis. Os testes foram realizados em ambiente simulado e os resultados mostram uma economia de até 50% de energia para um cenário com 10 UEs.

Em [16] é proposto o ORAN-HAutoscaling, um *framework* de escalonamento dinâmico para gerenciar eficientemente o uso de recursos computacionais em RICs. A partir de experimentos com *xApps* implantados em *clusters* Kubernetes, os autores evidenciam que o aumento do número de *xApps* pode levar à violação de requisitos de latência. Para mitigar esse problema, o *framework* adota estratégias de escalonamento horizontal e balanceamento de carga baseadas em métricas de uso de CPU, latência e tempo de inferência de modelos de Inteligência Artificial (IA). Os resultados demonstram que o ORAN-HAutoscaling permite suportar até três vezes mais *xApps* do que abordagens tradicionais, mantendo os limites de latência e otimizando a utilização de recursos, o que o torna uma solução promissora para garantir escalabilidade e desempenho em ambientes O-RAN.

No trabalho [17], é apresentado um método de predição de tráfego em redes sem fio no contexto de redes O-RAN, utilizando redes neurais. A proposta geral do sistema é baseada em um *xApp* que prevê o tráfego da rede e, com base em dois limiares dinâmicos de tráfego (alto e baixo), um orquestrador decide a inicialização de *xApps/rApps* voltados para suportar altas demandas ou para otimizar o consumo de energia. Os autores realizaram experimentos simulados e, com o sistema preditivo, conseguiram aumentar a eficiência energética média em 39,7% e a taxa de dados em 10,1%, em comparação ao uso contínuo de *xApps* específicos para cenários de alta e baixa demanda de tráfego, respectivamente.

Em [18], é proposto o conceito de cooperação entre *xApp* e *rApp* para a formulação de *clusters* de serviço, com foco na eficiência energética. Na proposta, o *rApp* é responsável por criar as políticas de formação dos *clusters* de serviço, como o número de unidades de rádio que devem compor cada *cluster*. O *xApp*, por sua vez, ajusta os *clusters* conforme as políticas recebidas, por meio da interface A1, garantindo a implementação adequada das diretrizes estabelecidas. Experimentos simulados demonstraram um ganho de até 37% em eficiência energética, em comparação com o estado da arte.

No estudo desenvolvido em [19], é proposto um orquestrador de *xApps*, projetado para maximizar os serviços oferecidos e diminuir o consumo de recursos. A ideia principal é que serviços podem compartilhar *xApps* sempre que estes corresponderem a funções semanticamente equivalentes, e a saída do *xApp* for de qualidade suficiente para atender aos requisitos do serviço. Resultados mostraram que a solução com o orquestrador supera o estado da arte ao implantar até 35% mais serviços com uma média de 30% menos *xApps* e uma redução similar no consumo de recursos.

O trabalho [20] apresenta um mecanismo de orquestração baseado em aprendizado por reforço hierárquico. O *xApp* desenvolvido é capaz de orquestrar múltiplos *rApps* e *xApps* de acordo com a intenção do operador de otimizar determinados KPIs, sendo estruturado em duas partes: um meta-controlador, que define as metas de desempenho em termos de KPIs; e um controlador, responsável pela orquestração em baixo nível. Resultados de testes simulados mostraram um aumento de até 21,4% na taxa média do sistema e uma melhoria de até 37,9% na eficiência energética, em comparação com outros *xApps* utilizados como *baseline*.

Por fim, os trabalhos apresentados, com exceção do estudo [14], propõem soluções voltadas à melhoria na eficiência energética no gerenciamento de recursos de rádio e na orquestração, mas não abordam especificamente a coleta de métricas. No entanto, independentemente da aplicação, o monitoramento de KPIs da rede é sempre necessário, pois é essencial obter informações sobre o desempenho da rede antes de tomar qualquer ação sobre ela. Nesse contexto, o *xApp* adaptativo proposto nesta dissertação visa otimizar o consumo energético no monitoramento de KPIs, podendo ser integrado às soluções apresentadas nos trabalhos anteriores para ampliar a economia de recursos. O mesmo raciocínio se aplica ao trabalho [14], que, embora já proponha uma abordagem para reduzir o consumo energético no monitoramento de KPIs, também pode ser complementado pela solução aqui desenvolvida.

1.3 Organização da Dissertação

Este trabalho está organizado da seguinte maneira:

- O Capítulo 2 fornece o referencial teórico necessário para a compreensão do trabalho;
- O Capítulo 3 descreve o funcionamento e o desenvolvimento do *xApp* proposto;
- O Capítulo 4 apresenta a metodologia adotada para a prova de conceito e o *setup* de medição, além de apresentar o cenário de investigação;
- No Capítulo 5 são apresentados e discutidos os resultados obtidos com o *xApp* proposto;
- Por fim, o Capítulo 6 apresenta as conclusões e a produção acadêmica do discente durante o mestrado.

Capítulo 2

Redes *Open* RAN

A arquitetura tradicional da RAN, composta por equipamentos monolíticos de fornecedores únicos, impõe restrições à flexibilidade da rede e impede a integração eficaz de novos recursos e estratégias de gerenciamento. Em resposta a essas limitações, surgiu o conceito de *Open* RAN, que propõe uma rede desagregada e orientada por *software*, conectada por interfaces abertas e interoperáveis. Essa mudança permite maior flexibilidade na composição da rede, incentiva a entrada de novos fornecedores e viabiliza que seu controle inteligente seja implementado por terceiros por meio de *xApps* e *rApps* [21].

2.1 Arquitetura de Referência da O-RAN Alliance

Fundada em 2018, a O-RAN Alliance é uma organização global composta por operadoras de telecomunicações, fabricantes de equipamentos, empresas de tecnologia e institutos de pesquisa. Seu principal objetivo é estabelecer um padrão aberto para a RAN e liderar a transição das redes para a arquitetura *Open* RAN. A organização fundamenta-se em dois pilares principais: interoperabilidade e inteligência. Para alcançar esses objetivos, a O-RAN Alliance desenvolveu modelos de referência, conhecidos como a visão O-RAN, que são alinhados à arquitetura definida pelo 3GPP e servem como base para a construção de redes abertas e inteligentes [8, 10].

Comumente chamada de *gNodeB* ou simplesmente gNB, a RAN definida pelo 3GPP para o 5G, na verdade, é formada pelos componentes: *Central Unit* (CU), *Distributed Unit* (DU) e *Radio Unit* (RU), como apresentado na Figura 2.1. Esses componentes são responsáveis por realizar diversas funções da pilha de protocolos associadas às camadas *Radio Resource Control* (RRC), *Non-Access-Stratum* (NAS), *Packet Data Unit* (PDU), *Service Data Adaption Protocol* (SDAP), *Packet Data Convergence Protocol* (PDCP), *Radio Link Control* (RLC), *Medium Access Control* (MAC) e *Physical* (PHY) [22].

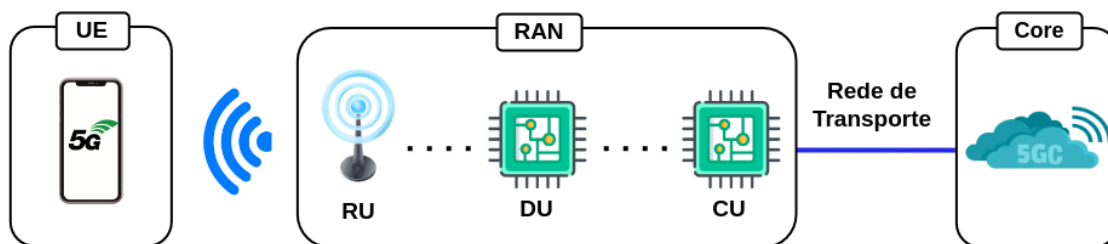


Figura 2.1: Arquitetura 5G RAN com split CU/DU.

Apesar da separação funcional entre RU, CU e DU trazer ganhos em termos de flexibilidade e escalabilidade, a interoperabilidade entre dispositivos de diferentes fabricantes continua sendo um desafio. Estes dispositivos, baseados em *hardware* especializado e *software* proprietário, seguem padrões distintos, o que limita a interoperabilidade. Dessa forma, a RAN é tradicionalmente composta por equipamentos de um único fornecedor, como ilustrado na Figura 2.2.

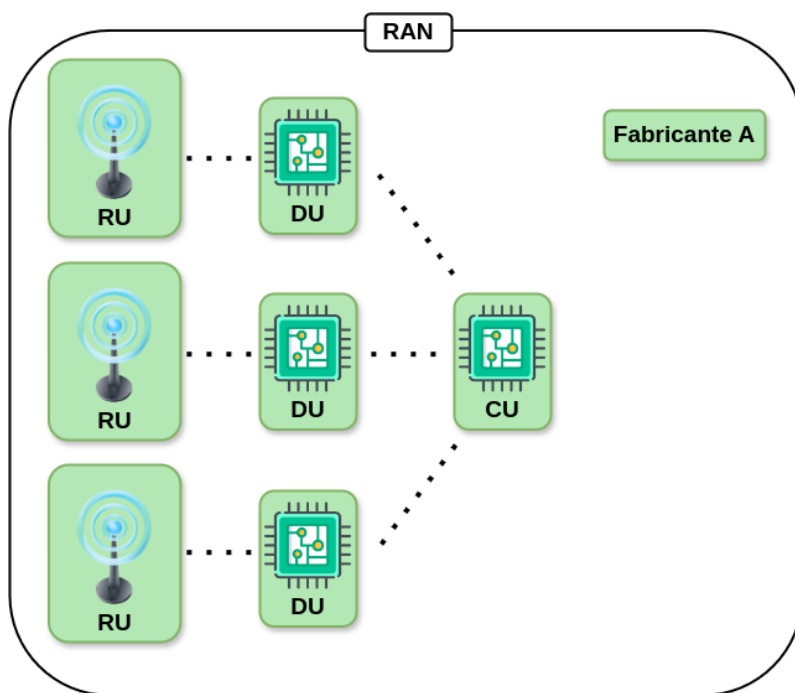


Figura 2.2: RAN proprietária.

Para promover um de seus pilares, a O-RAN propõe uma arquitetura de referência para a RAN baseada em interfaces abertas e na desagregação de *hardware* e *software*. Esse novo paradigma cria um ambiente competitivo, dinâmico e propício à inovação. Nele, as interfaces abertas garantem liberdade aos operadores na escolha dos componentes da rede, independentemente do fabricante. Além disso, a desagregação possibilita o uso de processadores de propósito geral, que têm menor custo em comparação com *hardwares* especializados [23, 24]. A Figura 2.3 ilustra essa arquitetura, destacando os componentes com o prefixo "O-", que indica compatibilidade com o padrão da O-RAN Alliance.

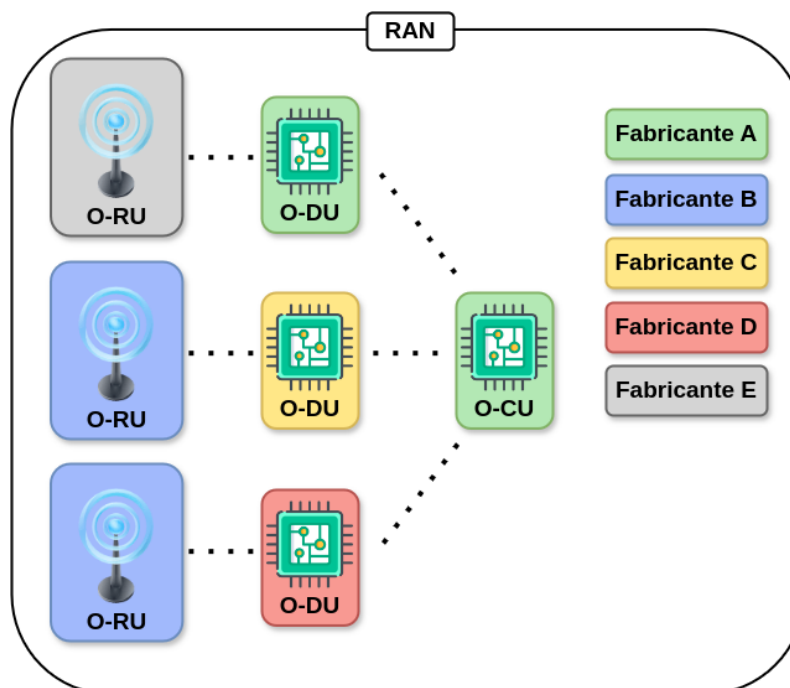


Figura 2.3: RAN aberta.

Para completar sua arquitetura de referência, a O-RAN *Alliance* promove outro pilar essencial: aprimorar a RAN com a incorporação de recursos inteligentes. Para esse fim, três elementos principais são definidos: o *Service Management and Orchestration* (SMO), dotado do *Non-RT RIC*; o *Near-RT RIC* e a infraestrutura O-Cloud. Esses elementos estão interligados por interfaces bem definidas e abertas, como A1, O1, O2 e E2. Os componentes O-CU, O-DU e O-RU, por serem controlados via interface E2, são chamados de nós E2. A Figura 2.4 apresenta uma visão lógica simplificada dessa arquitetura com esses componentes e suas principais interfaces, que serão detalhados a seguir:

- O **SMO** é responsável por orquestrar e gerenciar os recursos da RAN. Abriga o **Non-RT RIC**, uma função lógica com o objetivo de otimizar a RAN por meio de processos que não requerem ou não podem ser executados em tempo real. Por exemplo, pode ser responsável por gerenciar modelos de aprendizado de máquina e fornecer orientações à RAN baseadas em políticas de gerenciamento, utilizando análises de dados [25, 26]. Isso é viabilizado por meio de três interfaces principais, como mostrado na Figura 2.4:
 - **Interface A1:** utilizada para a otimização da RAN, conecta o *Non-RT RIC* ao *Near-RT RIC*. Sua função é transmitir políticas ao *Near-RT RIC*, possibilitando ajustes tanto no comportamento geral dos nós E2, como O-CU, O-DU, O-RU, quanto no próprio funcionamento do *Near-RT RIC* e de seus *xApps* [23, 2];
 - **Interface O1:** interliga o SMO às funções de rede O-RAN (*Near-RT RIC* e nós E2), visando garantir a operação eficiente e confiável da rede.

Suas funções incluem receber notificações de falhas, realizar alterações de configuração e coletar dados de desempenho dos nós E2 e *Near-RT RIC* [23, 2];

- **Interface O2:** interliga o SMO e a **O-Cloud**, permitindo que o SMO receba recursos de computação, armazenamento e rede. Além de receber dados de desempenho de infraestrutura e implantação, gerenciamento de configuração e falhas [23, 2].
- O **Near-RT RIC** é uma função lógica que permite o controle e a otimização quase em tempo real dos elementos e recursos da RAN, realizando decisões com latências da ordem de 10ms a 1s. Para isso, o *Near-RT RIC* hospeda os *xApps* que interagem com os nós E2 por meio da **interface E2**, utilizando o protocolo E2AP.
 - O **xApp** é um aplicativo modular executado no ambiente do *Near-RT RIC*, desenvolvido para realizar uma função específica de controle ou monitoramento da RAN. Um *xApp* pode, por exemplo, realizar balanceamento de carga, controle de interferência, alocação de espectro ou monitoramento de KPIs. Esses aplicativos operam em ciclos curtos e tomam decisões baseadas em dados recebidos da rede, em tempo quase real. A modularidade dos *xApps* permite que operadoras e terceiros desenvolvam e implementem novas funcionalidades de forma ágil, sem depender da atualização de toda a infraestrutura. A lógica de cada *xApp* pode ser atualizada dinamicamente, e múltiplos *xApps* podem coexistir, cada um com uma função específica, colaborando para a inteligência distribuída da RAN [2, 27].

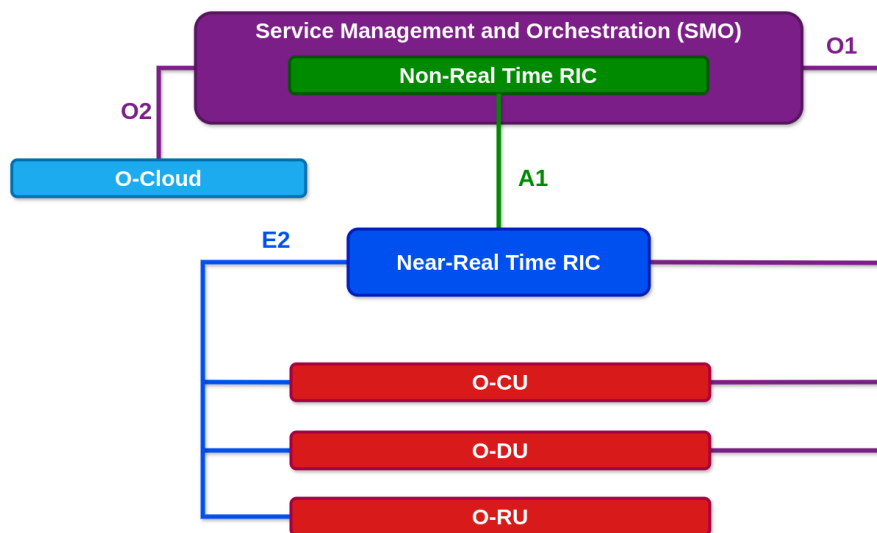


Figura 2.4: Arquitetura lógica simplificada da O-RAN [2].

2.2 Sobrecarga de Monitoramento via Interface E2

O monitoramento de métricas da RAN por meio da interface E2 desempenha papel central na viabilização de funcionalidades inteligentes no *Near-RT RIC*, especialmente por meio de *xApps* voltados à otimização de recursos e ao controle adaptativo da rede. Contudo, essa capacidade vem acompanhada de desafios computacionais e energéticos significativos, principalmente em cenários de larga escala [14, 16].

Experimentos realizados com tráfego real e medições de consumo de energia demonstraram que tanto o volume de tráfego na interface E2 quanto o número de KPIs reportados por nó estão linearmente relacionados ao consumo energético do RIC. Esses resultados evidenciam que a coleta contínua de KPIs pode se tornar um gargalo para a eficiência energética dos controladores inteligentes da RAN [14].

Além disso, os *xApps* realizam assinaturas periódicas para receber conjuntos de KPIs definidos pelo padrão da O-RAN *Alliance*, os quais abrangem diversas camadas da pilha de protocolos, como exemplificado a seguir:

- KPIs da camada PHY:
 - SS-RSRP (DL *Synchronization Signal based Reference Signal Received Power*);
 - SRS-RSRP (UL *Sounding Reference Signal based Reference Signal Received Power*);
 - RSRQ (DL *Reference Signal Received Quality*).
- KPIs da camada MAC:
 - TBS (*Transport Block Size*) agregado de *Downlink* e *Uplink*;
 - SNR do PUSCH (*Physical Uplink Shared Channel*);
 - SNR do PUCCH (*Physical Uplink-Control Channel*);
 - CQI (*Channel Quality Indicator*);
 - MCS (*Modulation Coding Scheme*);
 - BLER (*Block Error Rate*);
 - *Power Headroom Report* (PHR).
- KPIs da camada RLC:
 - Pacotes ou bytes de PDU transmitidos;
 - Pacotes ou bytes de PDU retransmitidos;
 - Ocupação atual do buffer de transmissão (em pacotes ou bytes);
 - Bytes recebidos pelos PDUs da RLC;
 - Número de pacotes recebidos descartados ou ignorados pela RLC.
- KPIs da camada PDCP:
 - Pacotes ou bytes de PDU transmitidos;
 - Pacotes ou bytes de PDU recebidos;
 - Número de pacotes ou bytes recebidos fora de ordem;
 - Número de pacotes ou bytes descartados;
 - Número de SDUs recebidos.

- KPIs da camada RRC:
 - Temporizador de inatividade do RRC;
 - Solicitação de conexão RRC;
 - Informações de UE em *Sidelink*.

O documento [28] define formalmente os KPIs que podem ser trocados entre o *Near-RT RIC* e as *gNBs* do sistema celular, tanto sob a perspectiva da estação rádio base quanto dos equipamentos de usuário (UEs). Essa abrangência amplia o volume de sinalização pela interface E2, elevando o consumo de recursos computacionais e de rede, especialmente quando os intervalos de amostragem dos KPIs são curtos.

A sobrecarga gerada por essa coleta intensiva de métricas torna-se ainda mais crítica diante do crescente uso de algoritmos baseados em *big data* e *deep learning* no RIC, que frequentemente demandam grandes volumes de dados em tempo quase real para tomada de decisão.

Também é importante mencionar que a interface E2 é utilizada pelo RIC para enviar comandos de controle para os nós E2, geralmente em resposta a alguma criticidade de desempenho ou uso de recursos previamente reportados. Assim, a interface E2 é responsável por trafegar fluxo em duas direções, da *Next Generation Node B* (*gNB*) para o RIC e vice-versa.

Diante desse contexto, o Capítulo 3 apresenta o desenvolvimento de um *xApp* voltado ao monitoramento adaptativo de KPIs, com o objetivo de mitigar o uso excessivo de recursos decorrente da coleta contínua de métricas. A proposta fundamenta-se em uma metodologia capaz de ajustar dinamicamente a frequência de amostragem, buscando equilibrar o consumo de recursos com a qualidade da informação reportada ao RIC.

Capítulo 3

Solução Proposta: *xApp* adaptativo

Com o objetivo de reduzir o consumo de CPU, recursos energético e de rede, este trabalho propõe um *xApp* adaptativo que utiliza uma frequência de amostragem variável ao longo do processo de monitoramento. O *xApp* proposto utiliza uma solução que ajusta dinamicamente o tempo de amostragem de KPIs que trafegam na interface E2, alterando a granularidade temporal da coleta com base em uma **avaliação de risco**. Para isso, o seu funcionamento foi estruturado em duas fases, as quais serão detalhadas neste capítulo.

A primeira fase do *xApp*, chamada de **KPI Evaluation Phase (KEP)**, tem como objetivo definir um nível de risco associado ao KPI monitorado. Inicialmente, realiza-se a coleta das amostras com alta frequência, utilizando um tempo de amostragem fixo de $t_s = 1\text{ ms}$, conforme ilustrado no bloco verde da Figura 3.1. Essa coleta gera um vetor $\mathbf{x} = [x_1, x_2, \dots, x_n]$ contendo n amostras sucessivas do KPI, que é então encaminhado à etapa de avaliação de risco (blocos em vermelho).

Nesta etapa, busca-se quantificar a variabilidade do KPI ao longo da janela de observação. Para isso, calcula-se inicialmente o vetor de diferenças entre amostras consecutivas:

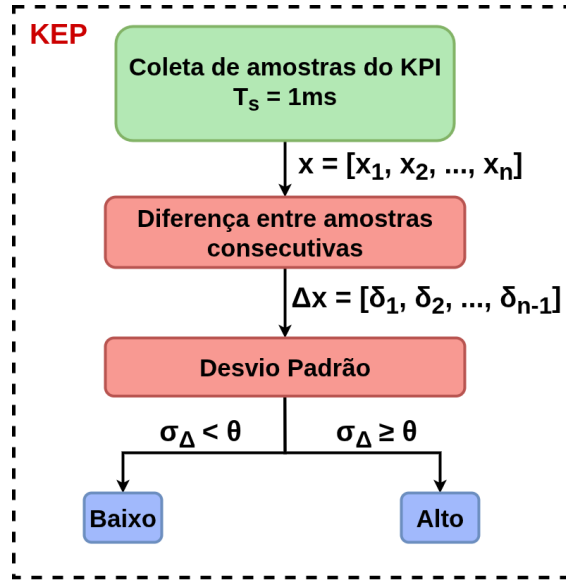
$$\Delta \mathbf{x} = [\delta_1, \delta_2, \dots, \delta_{n-1}], \quad \text{em que} \quad \delta_i = x_{i+1} - x_i, \quad \forall i = 1, \dots, n-1. \quad (3.1)$$

Em seguida, computa-se o desvio padrão dessas variações, denotado por σ_Δ , como medida da dispersão das diferenças em torno da média $\bar{\delta}$:

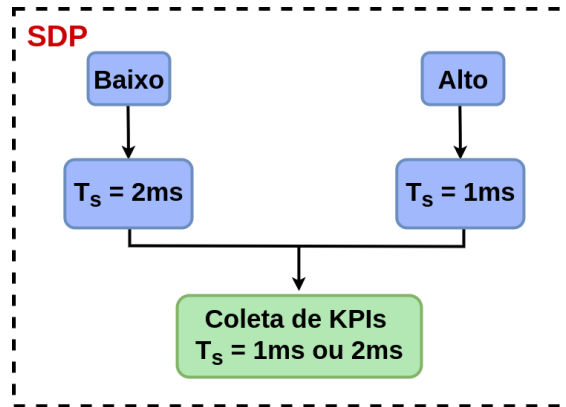
$$\sigma_\Delta = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (\delta_i - \bar{\delta})^2}. \quad (3.2)$$

O valor de σ_Δ representa a intensidade das oscilações do KPI na janela atual e é utilizado como critério para a classificação do risco. Por fim, define-se um limiar θ , previamente calibrado, e a classificação ocorre da seguinte forma:

$$\text{Risco} = \begin{cases} \text{Baixo,} & \text{se } \sigma_\Delta < \theta \\ \text{Alto,} & \text{se } \sigma_\Delta \geq \theta \end{cases} \quad (3.3)$$

Figura 3.1: Diagrama de bloco da fase *KPI Evaluation Phase* (KEP).

Na segunda fase do *xApp*, chamada de *Sampling Time Definition Phase* (SDP), é realizada a definição do tempo de amostragem a ser adotado na próxima janela de coleta, como apresentado na Figura 3.2. Essa decisão baseia-se diretamente no nível de risco identificado na fase KEP. Caso o risco seja classificado como alto, o *xApp* mantém o tempo de amostragem em $t_s = 1\text{ ms}$, preservando a granularidade para monitoramento mais preciso. Por outro lado, se o risco for considerado baixo, a frequência de coleta é reduzida com o objetivo de otimizar o uso de recursos. Nessa situação, o tempo de amostragem é ajustado para $t_s = 2\text{ ms}$, promovendo uma coleta mais esparsa.

Figura 3.2: Diagrama de bloco da fase *Sampling Time Definition Phase* (SDP).

O funcionamento do *xApp* adaptativo é apresentado na Figura 3.3, que ilustra o diagrama de blocos completo, composto pelas fases KEP e SDP. Como apresentado anteriormente, existem duas etapas de coleta de KPIs, uma com alta frequência, na fase KEP e outra com frequência variável, na fase SDP. A soma do tempo gasto nessas duas fases define a duração do *Duty-cycle* do *xApp*, sendo, portanto, estabelecidas

duas variáveis temporais que interferem diretamente em seu desempenho e fornecem flexibilidade à sua configuração:

- **DC**: tempo de duração do *Duty-cycle*, em segundos;
- **Alpha** (α): tempo de duração da coleta na fase KEP, também em segundos.

A escolha desses dois parâmetros irá ditar o comportamento do *xApp* ao longo de sua execução. Supondo que $DC = 1\text{ s}$ e $\alpha = 0,5\text{ s}$, e considerando que o *xApp* realizará o monitoramento por um intervalo total de 1 minuto, observa-se que o *Duty-cycle* será repetido 60 vezes nesse período, ou seja, a cada 1 segundo será tomada a decisão de manter ou diminuir a frequência de amostragem. Além disso, como α corresponde à metade do valor de DC , infere-se que as fases KEP e SDP ocupam, cada uma, 50% do tempo de cada ciclo.

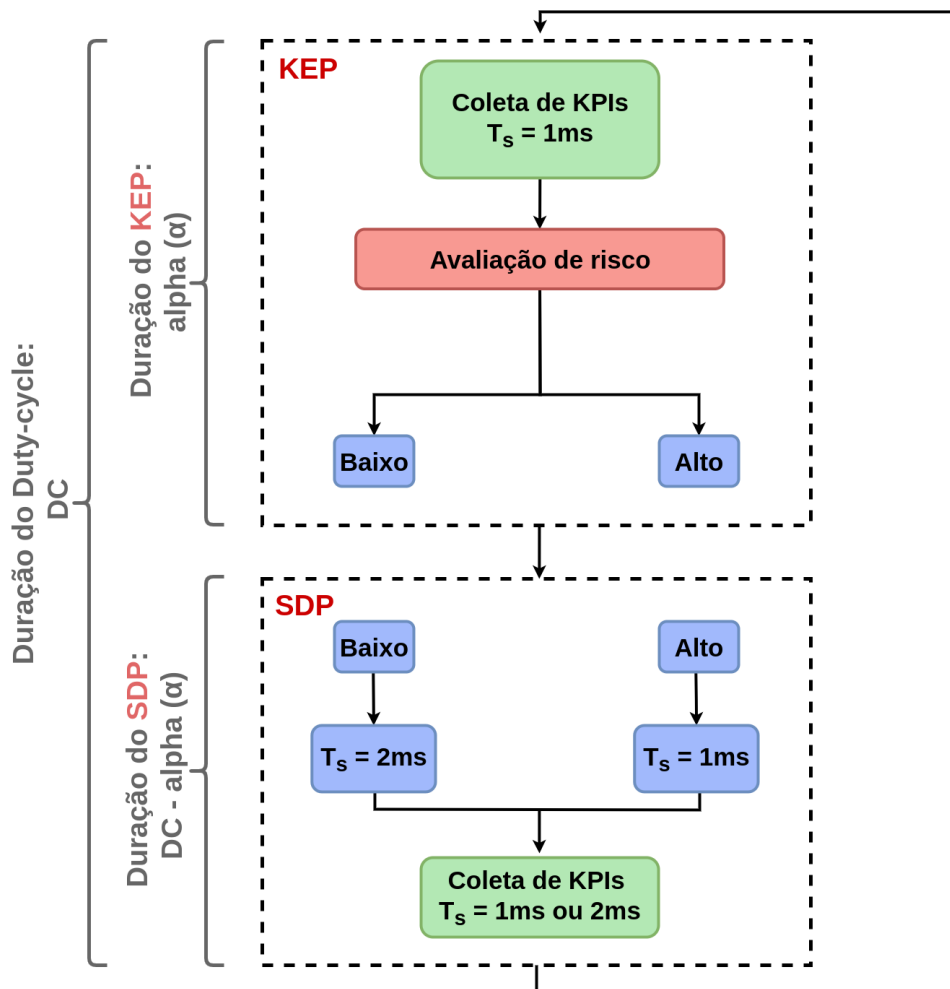


Figura 3.3: Diagrama de bloco do xApp Adaptativo.

Nesse sentido, a escolha do DC define, indiretamente, a quantidade de *Duty-cycles* que ocorrem em um determinado intervalo de tempo de monitoramento, enquanto o valor

de α determina a fração de cada *Duty-cycle* dedicada à fase KEP. A fase SDP, por sua vez, é executada a cada *Duty-cycle*, com duração igual a $DC - \alpha$. A Figura 3.4 ilustra três variações de composição entre DC e α , bem como suas distribuições ao longo do tempo. O primeiro caso ilustra o exemplo comentado com $DC = 1\text{ s}$ e $\alpha = 0,5\text{ s}$, em que a distribuição das fases KEP e SDP é igual e unitária a cada *Duty-cycle*. Na segunda situação, ocorre a diminuição do tempo α , em que as duas fases continuam ocorrendo apenas uma vez a cada *Duty-cycle*, contudo com predominância da fase SDP. Já no último caso, observa-se que a definição de $DC = 0,25\text{ s}$ e $\alpha = 0,25\text{ s}$ resulta no dobro de execuções das fases KEP e SDP em comparação com os demais cenários.

Em relação ao primeiro e ao último caso, percebe-se que são semelhantes no que se refere ao tempo total das fases KEP e SDP, uma vez que, ao se analisar um segundo de medição, observa-se a ocupação de 0,5 s por essas duas fases. No entanto, tratam-se de situações distintas: no primeiro caso, realiza-se apenas uma avaliação de risco por segundo, enquanto no último são realizadas duas. Assim, embora ambos os casos apresentem duração total equivalente, o último oferece uma maior resolução na avaliação de risco.

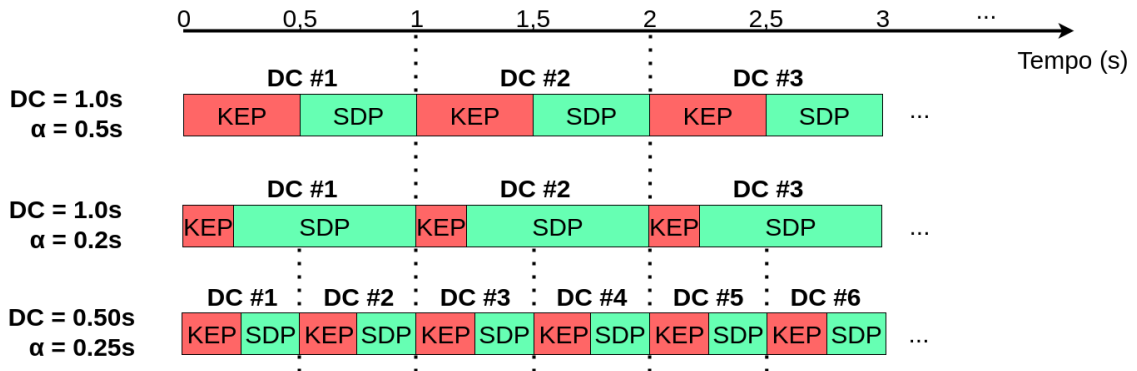


Figura 3.4: Composição de KEP e SDP no *Duty-cycle*.

Por fim, é importante destacar que os valores de DC e α , a definição da Avaliação de Risco e as opções de tempo de amostragem podem ser modificados, gerando variantes do *xApp* proposto. Portanto, os valores escolhidos para essas variáveis, bem como o algoritmo da Avaliação de Risco, fazem parte da prova de conceito avaliada nesta dissertação.

3.1 Implementação do *xApp* adaptativo

A implementação da solução proposta foi desenvolvida sobre a plataforma *Near-RT RIC* do *OpenAirInterface* (OAI), chamada de *FlexRIC*. Ela permite o uso e o desenvolvimento de *xApps* compatíveis com a arquitetura definida pela *O-RAN Alliance*. O *FlexRIC* fornece um exemplo de *xApp* voltado para o monitoramento de métricas da RAN, sendo esse o ponto de partida para a implementação do *xApp* adaptativo. O mesmo

é concebido sob uma arquitetura híbrida, combinando a linguagem Python para a lógica de controle de alto nível com o alto desempenho da linguagem C para a coleta de dados.

Alguns desafios foram enfrentados para se chegar na implementação do *xApp* adaptativo a partir do exemplo fornecido pelo FlexRIC, os quais são enumerados a seguir:

- Refatoração para modularizar o *xApp*. Justificativa: o exemplo fornecido pelo OAI consiste em um único *script* com fluxo de execução linear, o que dificulta a implementação da lógica de controle dinâmico do *xApp* adaptativo;
- Refatoração do código para monitorar apenas as métricas da camada MAC. Justificativa: o exemplo fornecido monitora métricas das camadas MAC, RLC e PDCP. Como os experimentos realizados neste trabalho geram um grande volume de dados, optou-se por restringir as camadas monitoradas, visando otimizar o processamento e o armazenamento de dados, bem como o tempo de execução dos testes de prova de conceito;
- Implementação do arcabouço para manipulação do banco de dados. Justificativa: o exemplo fornecido pelo OAI apenas salva as métricas no banco de dados. Assim, foi necessária a implementação de métodos para consulta e manipulação dos dados em tempo de execução, especialmente para utilizá-los na avaliação de risco da fase KEP;
- Alteração do código fonte para a utilização de diferentes tempos de amostragem. Justificativa: o código inicial permitia apenas quatro opções de tempo de amostragem, o que limitava os testes;
- Alteração do código fonte para modificar o intervalo de valores aceitos para a métrica PHR da camada MAC. Justificativa: o intervalo definido inicialmente é incompatível para medições com UE real, ocasionando falha na coleta de métricas ao executar o *xApp*;
- Alteração do código fonte para solucionar problema de latência nas medições. Justificativa: durante os experimentos percebeu-se que as medições duravam mais tempo do que o definido. Assim, a partir de uma investigação no código fonte, identificou-se que chamadas a funções do tipo *sleep()* ocasionavam atraso na coleta de métricas e, consequentemente, no tempo de medição. O problema foi solucionado ao remover essas funções.

Após as alterações necessárias para a implementação do *xApp* adaptativo, seu funcionamento final foi definido como apresentado na Figura 3.5. O *xApp*, implementado na linguagem Python, inicia seu funcionamento na Etapa 1 (ilustrada na Figura 3.5) ao acionar a fase KEP, chamando a função de coleta de métricas com tempo de amostragem $t_s = 1\text{ ms}$ e tempo de coleta α , definidos pelo operador. A função de coleta de métricas é mapeada para uma função escrita na linguagem C, por meio do *wrapper* SWIG. A coleta de métricas é então realizada pela função em C, a qual recebe amostras enviadas pelo nó E2, durante a duração definida em α , e as armazena em um banco de dados SQLite (Etapa 2).

Em seguida, as amostras são coletadas pelo código em Python na fase KEP (Etapa 3) e é realizada a análise de risco para definir o novo tempo de amostragem t_s (Etapa 4). Inicia-se assim a fase SDP com o novo tempo de amostragem definido na fase KEP,

sendo novamente chamada a função de coleta de métricas em C (Etapa 5), a qual irá receber amostras do nó E2 durante o tempo $DC - \alpha$. Por fim, os dados da fase SDP são armazenados no banco SQLite (Etapa 6) e o ciclo é reiniciado.

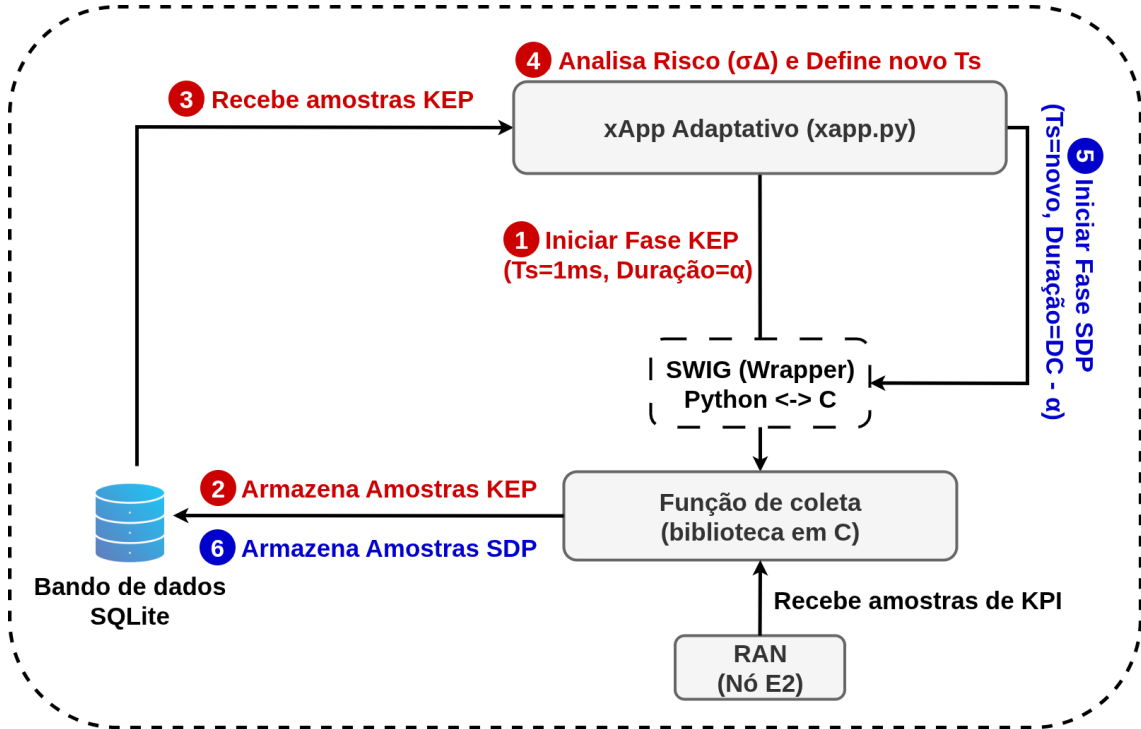


Figura 3.5: Diagrama de fluxo do algoritmo.

O próximo capítulo apresenta a metodologia de avaliação de prova de conceito do *xApp* proposto.

Capítulo 4

Metodologia de Prova de Conceito

Neste capítulo, apresenta-se a metodologia de investigação adotada, detalhando cada componente do *setup* de medição, assim como o procedimento utilizado para comparar o *xApp* de monitoramento tradicional (com frequência de amostragem fixa) e o *xApp* de monitoramento adaptativo proposto neste estudo.

4.1 *Setup* e Cenário de Medição

O *setup* de medição utilizado neste trabalho foi concebido para avaliar o desempenho de *xApps* de monitoramento no contexto de redes *Open* RAN 5G. Para isso, foi configurada uma rede 5G fim-a-fim, composta pelo núcleo de rede, RAN e UE. Além disso, para suportar o desenvolvimento e a execução de *xApps*, foi integrado à rede um RIC.

Para a implantação do núcleo de rede e da RAN foi utilizado o *software* de código aberto OAI, o qual é mantido por uma comunidade internacional de desenvolvedores. O OAI oferece soluções para RAN e núcleo de rede, com foco em tecnologias 4G e 5G, cobrindo toda a pilha de protocolos definidos pelo 3GPP. Ele permite a construção e personalização dessas redes com suporte tanto para plataformas de Rádio definido por *Software* (SDR) quanto para RUs comerciais. Ao utilizar *front-ends* de RF (SDRs ou RUs), o código permite a comunicação da gNB com UEs comerciais [29, 30]. Além disso, o OAI integra uma implementação do *Near-RT* RIC compatível com a arquitetura da O-RAN: o *software FlexRIC* [31], citado na Seção 3.1.

A rede 5G implementada utiliza o núcleo de rede na versão v1.5.0, disponível em [32], e a RAN na versão 2023.w37, disponível em [33]. Para o *FlexRIC*, foi necessário modificar o código-fonte com as alterações mencionadas na Seção 3.1, sendo versionado em repositório próprio privado (disponível em <https://projetos.imd.ufrn.br/reginalab/nric-flexric/-/tree/master-init>). Todos os componentes foram instalados em um *desktop* Lenovo ThinkCentre N50s Gen 3, com as seguintes especificações:

- Sistema Operacional: Linux 20.04 LTS;
- Processador: Intel(R) Core(TM) i7-12700, 12 núcleos, 2,10 GHz;
- Memória RAM: 64 GB.

Para o funcionamento da gNB, a frequência da portadora foi definida na banda n78, especificamente entre 3,7 e 3,8 GHz, pois o grupo de pesquisa que dá suporte a este trabalho tem permissão da ANATEL para utilizá-la para fins científicos [34]. Além disso, foi associada ao *desktop* uma plataforma SDR, modelo USRP Ettus B210¹, equipada com duas antenas Ettus LP0965², com as seguintes especificações:

- Modelo: Log-periódica;
- Diagrama de Radiação: Direcional;
- Ganho: 5 - 6 dBi;
- Faixa de Frequências: 850 - 6500 MHz.

Em relação ao UE, foi utilizado um dispositivo Motorola G50 5G, um equipamento *Commercial Off-The-Shelf* (COTS), equipado com um *Universal Subscriber Identity Module* (USIM), da Sysmocom. Foi utilizado um gravador de USIM para configurar o *simcard* para conexão com o núcleo da rede do OAI instalado. Assim, o *setup* fim-a-fim completo é apresentado na Figura 4.1.

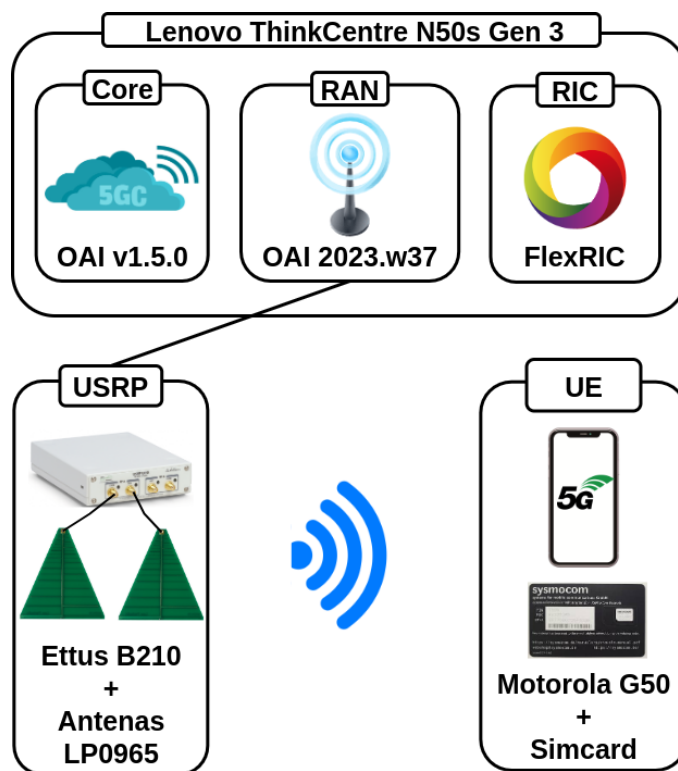


Figura 4.1: Setup de medição.

No que se refere à coleta de métricas, ela ocorre por meio de um *xApp* implementado no FlexRIC. A função de monitoramento de KPIs depende de três componentes distintos, cada um com responsabilidades específicas:

¹USRP: <https://www.ettus.com/all-products/ub210-kit/>

²Antenas: <https://www.ettus.com/all-products/lp0965/>

- **xApp de monitoramento:** aplicação de *software* que contém a lógica em código para monitorar a RAN, definindo quais KPIs devem ser coletados, com que frequência e como esses dados devem ser processados após recebidos;
- **FlexRIC:** componente central que gerencia as solicitações dos xApps e estabelece e mantém a comunicação com os elementos da RAN (nó E2);
- **Agente do nó E2:** módulo instalado na RAN (OAI), responsável por expor as funcionalidades internas e os dados da RAN para o FlexRIC, por meio da interface E2.

A comunicação entre esses três componentes é estabelecida pelas interfaces E42 (entre o xApp de monitoramento e o FlexRIC) e a E2 (entre o FlexRIC e o agente do nó E2). A Figura 4.2 ilustra a troca de mensagens que viabiliza o monitoramento das métricas. Primeiramente, o xApp inicia a comunicação com o FlexRIC, por meio da mensagem **E42 Setup Request**, e recebe em resposta a **E42 Setup Response**, confirmando que a conexão foi estabelecida com sucesso. Em seguida, por meio da **E42 Subscription Request**, o xApp solicita a subscrição, informando quais métricas devem ser coletadas, de qual nó E2 e com que frequência. O FlexRIC recebe essa solicitação e extrai dela a mensagem **E2 Subscription Request**, que é encaminhada para o agente do nó E2. Este, por meio da **E2 Subscription Response**, confirma a aceitação da subscrição e o início do monitoramento conforme solicitado. Por fim, as métricas solicitadas passam a ser enviadas periodicamente ao FlexRIC na mensagem **E2 indication**, que as repassa ao xApp pela **E42 indication**, para que sejam salvas no banco de dados SQLite.

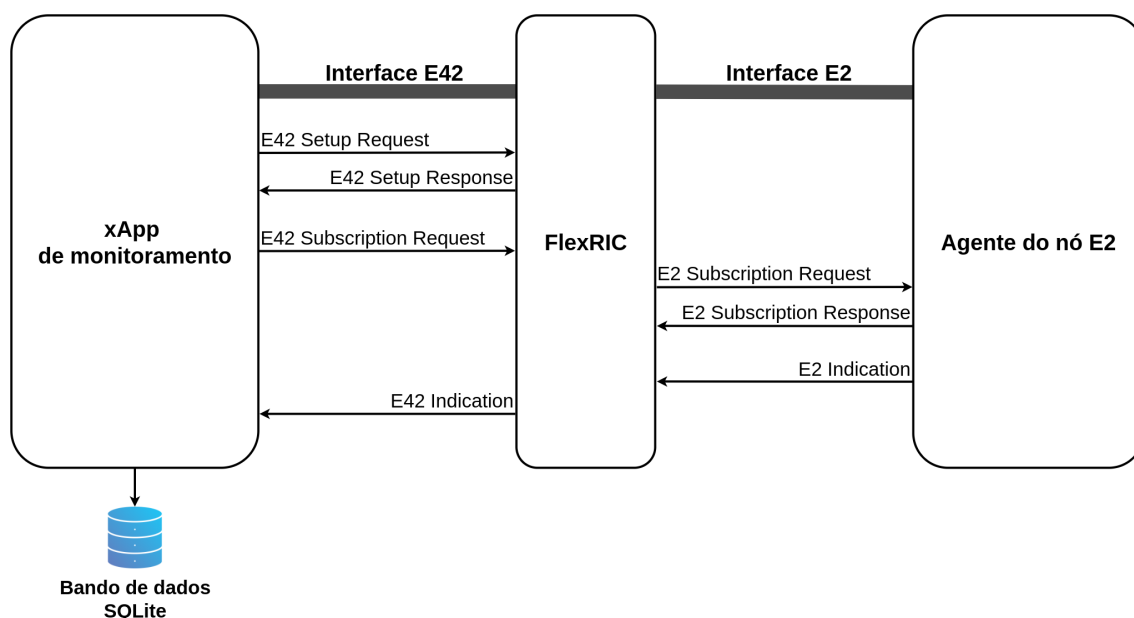


Figura 4.2: Fluxo de mensagens para coleta de métricas via xApp no FlexRIC.

Por outro lado, para o monitoramento do consumo de CPU, utilizou-se o *software* Zabbix, que é uma plataforma voltada para o monitoramento de desempenho e disponibilidade. O Zabbix é capaz de coletar, analisar e emitir alertas sobre métricas

de infraestrutura de TI, redes, servidores e aplicações em tempo real [35]. Seu funcionamento é baseado em um *Agent*, responsável por coletar os dados do sistema monitorado e enviá-los ao *Server*, que processa, armazena e analisa as informações, gerando alertas e relatórios.

Assim, o *Agent* foi instalado no *desktop* Lenovo ThinkCentre N50s Gen 3, o mesmo apresentado na Figura 4.1, e o *Server* em um servidor local. A Figura 4.3, apresenta o esquema de funcionamento do Zabbix no contexto abordado, em que a comunicação entre o *Agent* e o *Server* ocorre via rede local, utilizando a porta 10050 em modo passivo. Por meio dela, a cada 1 s, o *Server* realiza requisições ao *Agent* para obter os dados de interesse. O monitoramento ocorre em tempo real e o consumo de cada núcleo de processamento é apresentado graficamente via interface *web* padrão do Zabbix. Os valores são também armazenados em um banco de dados PostgreSQL, os quais são coletados via *script* para serem utilizados nas análises de pós-processamento.

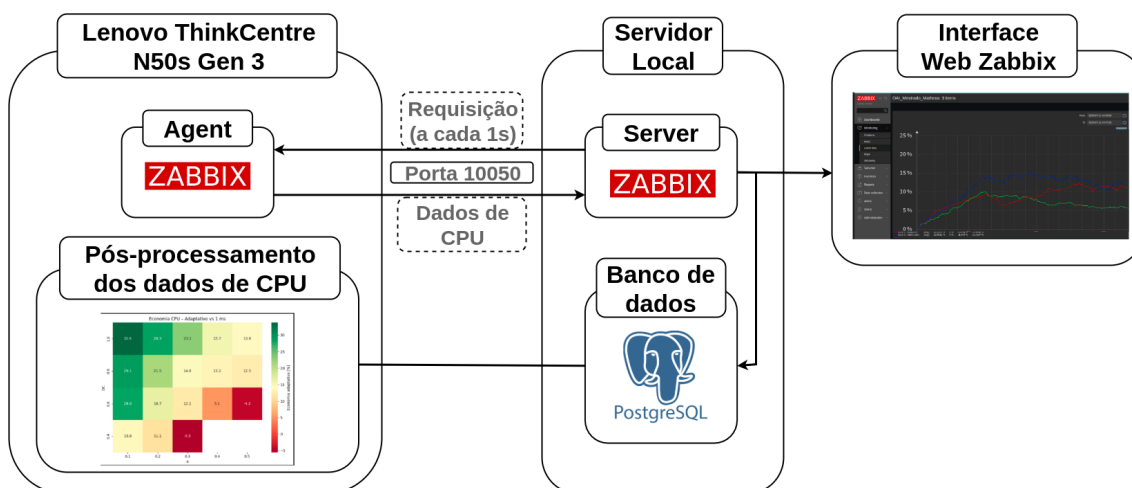


Figura 4.3: Sistema de monitoramento via Zabbix.

O *setup* apresentado foi implantado no centro de pesquisa *Leading Advanced Technologies Center of Excellence* (LANCE), situado no último andar do Núcleo de Pesquisa e Inovação em Tecnologia da Informação (nPITI) - IMD/UFRN, onde as medições foram realizadas. A Figura 4.4 ilustra o cenário de medição, no qual o UE foi posicionado no Laboratório GppCom/Regina, enquanto a gNB foi disposta no Laboratório Lenovo. Observa-se que o ambiente é caracterizado por um cenário rico em obstáculos, com a divisória entre os laboratórios composta por vidro e concreto, além da presença de estações de trabalho e armários constituídos de madeira e estruturas metálicas. Esses elementos contribuem para a formação de um ambiente com forte presença de multipercursos, desafiando a qualidade do enlace de rádio, e gerando KPIs que apresentem variação ao longo do tempo, mesmo que o celular permaneça parado.

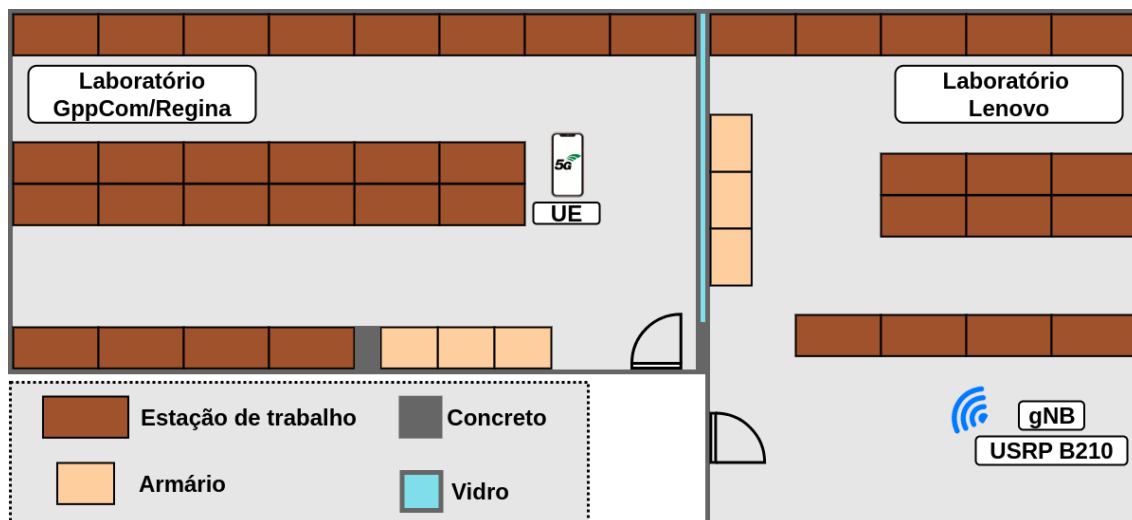


Figura 4.4: Cenário de medição.

Diante desse cenário, foram conduzidos dois procedimentos de medição distintos, com os seguintes objetivos:

1. Definição do limiar de decisão do algoritmo do *xApp* adaptativo;
2. Avaliação comparativa do consumo de CPU e da acurácia do *xApp* adaptativo e do *xApp* tradicional.

Apresentados na Seção 4.2, em ambos os experimentos, o UE executou uma transmissão ao vivo em 4K pelo YouTube, com o objetivo de gerar demanda por recursos da rede, permitindo assim avaliar o desempenho dos *xApps* em condições realistas de uso.

4.2 Procedimentos de Medição

Nesta seção, são apresentados dois procedimentos de medição distintos, denominados, respectivamente, **Definição do Limiar de Decisão** e **Avaliação de Desempenho**. O primeiro visa determinar um valor de referência para os riscos calculados na fase KEP, servindo como critério decisório para ajustes na frequência de amostragem. Já o segundo procedimento busca comparar o consumo de CPU e a acurácia do *xApp* tradicional e do *xApp* proposto, considerando diferentes configurações de parâmetros que influenciam o seu comportamento. A seguir, cada um desses procedimentos é detalhado.

Conforme ilustrado na Figura 3.1, com base no nível de risco calculado, a fase KEP é responsável por tomar decisões quanto à manutenção ou aumento do tempo de amostragem. Essa decisão é tomada a partir da comparação entre o risco estimado (σ_Δ) e um limiar de decisão (θ), previamente definido para o cenário em análise. Diante disso, o **Definição do Limiar de Decisão** tem como objetivo determinar esse limiar, considerando o ambiente descrito na Seção 4.1. O procedimento consiste em uma bateria de medições para coletar um conjunto de amostras de valores de risco calculados pela fase KEP. A Figura 4.5 apresenta a estrutura de repetição utilizada, que consiste na fase KEP

tradicional, mas a cada iteração em k , o valor de $\sigma_{\Delta}(k)$ é armazenado no vetor $\mathbf{v}\sigma_{\Delta}$. Ao final do experimento, esse vetor contém a série histórica de todos os valores assumidos por $\sigma_{\Delta}(k)$.

Finalmente, constrói-se a Função de Distribuição Acumulada (CDF) empírica com base nos valores de $\mathbf{v}\sigma_{\Delta}$. O limiar de decisão é então definido como o valor correspondente ao percentil de 50% da CDF, ou seja, o ponto no qual 50% dos valores observados de risco são inferiores ou iguais a ele. Essa escolha representa um ponto de corte robusto, pois representa a mediana da distribuição, a qual não é sensível à valores extremos; e balanceado, visto que reage de maneira equilibrada na decisão de risco alto ou baixo.

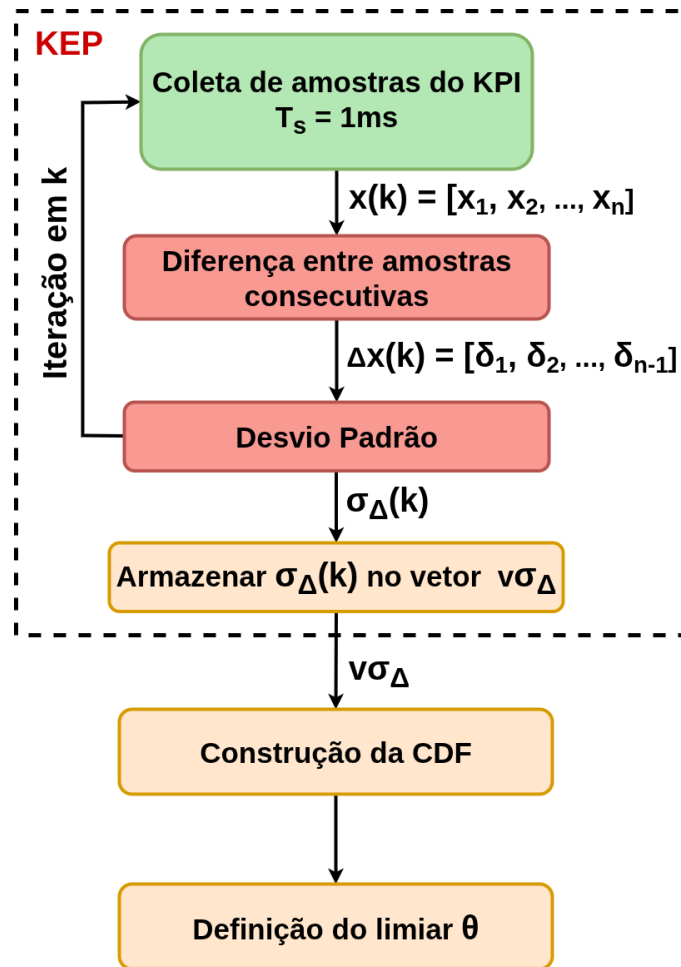


Figura 4.5: Diagrama da definição do limiar de decisão.

Concluído o primeiro procedimento, inicia-se a **Avaliação de Desempenho**, que é definida para fazer a análise comparativa do consumo de CPU e da acurácia do *xApp* adaptativo e do *xApp* tradicional estático. Para esse fim, são utilizados três *xApps*, a descrição de cada um é apresentada a seguir:

- ***xApp* adaptativo:** proposto neste trabalho, com frequência de amostragem adaptativa;

- ***xApp* estático - 1 ms**: tradicional com frequência de amostragem estática, configurado com **alta** taxa de amostragem (intervalo de 1 ms entre amostras);
- ***xApp* estático - 2 ms**: tradicional com frequência de amostragem estática, configurado com **baixa** taxa de amostragem (intervalo de 2 ms entre amostras).

Os *xApps* mencionados são executados simultaneamente durante as medições, realizando a coleta de KPIs da camada MAC. O ***xApp* estático - 1ms** é utilizado como referência para a análise comparativa, em que a partir do seu consumo de CPU é calculada a economia de CPU proporcionada pelo ***xApp* estático - 2ms** e pelo ***xApp* adaptativo**. Para manter as mesmas condições de medição, foi configurada a afinidade de CPU de modo que cada *xApp* fosse executado em um único núcleo de processamento, isolado e dedicado exclusivamente à sua execução. Isso garantiu que os três *xApps* fossem executados simultaneamente, sob as mesmas condições de rede sem fio.

Além disso, conforme discutido no Capítulo 3, os valores atribuídos aos parâmetros DC e α exercem forte influência sobre o comportamento do ***xApp* adaptativo**. Dessa forma, a sistemática de comparação do consumo de CPU foi aplicada em uma bateria de medições para diferentes combinações desses parâmetros. Então, foi construído um *grid* de medições, no qual cada ponto representa a economia de CPU obtida pelo ***xApp* adaptativo** para um determinado par de valores de DC e α . A Figura 4.6 apresenta um esboço desse *grid*, em que cada ponto do *grid* é realizada uma medição de 20 minutos. Vale ressaltar que os pontos em que $\alpha \geq DC$ não são considerados, uma vez que $\alpha = DC$ implica que a fase KEP ocuparia integralmente o *Duty-cycle*, não restando tempo para a execução da fase SDP. Para o caso em que $\alpha > DC$, trata-se de uma configuração inviável, pois a duração da fase KEP ultrapassaria o próprio *Duty-cycle*. Já o ***xApp* estático** não é influenciado pelos parâmetros DC e α e é executado simultaneamente ao ***xApp* adaptativo** para fins comparativos.

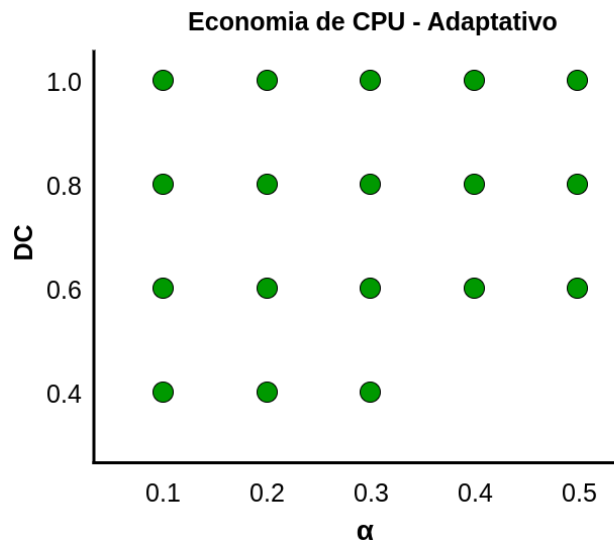


Figura 4.6: Esboço do *grid* de medição para diferentes valores de DC e α .

Além da avaliação de desempenho relacionada ao consumo de CPU, é realizada uma

segunda análise com o objetivo de comparar a acurácia dos *xApps* ao serem utilizados em uma aplicação real. Para essa avaliação, utilizam-se como base os bancos de dados contendo as métricas previamente coletadas pelos três *xApps*. A partir desses dados, é conduzido um pós-processamento com o intuito de simular o funcionamento de uma aplicação prática. Assim, foi desenvolvida uma simulação baseada na tomada de decisão do MCS, com base nos valores da métrica SNR do canal PUSCH.

A primeira etapa consiste na geração de uma tabela genérica de mapeamento entre valores de SNR e os respectivos índices de MCS. Para isso, são agregadas todas as amostras de SNR coletadas pelo *xApp* estático com tempo de amostragem de 1 ms. A partir da distribuição desse conjunto de dados, são definidos os limiares de SNR utilizados no mapeamento, obtidos por meio da extração de cinco pontos de corte baseados em percentis equidistantes da distribuição. Cada intervalo resultante é associado a um índice fixo de MCS, de modo que valores de SNR inferiores ao primeiro limiar correspondem ao MCS 0, enquanto os intervalos subsequentes são mapeados para os índices 5, 9, 14 e 17, respectivamente³. A Tabela 4.1 apresenta o índice de MCS associado a cada intervalo de SNR, definidos a partir dos percentis 20, 40, 60, 80 e 100% da distribuição, representados por P_{20} , P_{40} , P_{60} , P_{80} e P_{100} , respectivamente.

Tabela 4.1: Mapeamento entre intervalos de SNR e índices de MCS.

Intervalo de SNR	Índice de MCS
$\text{SNR} < P_{20}$	0
$P_{20} \leq \text{SNR} < P_{40}$	5
$P_{40} \leq \text{SNR} < P_{60}$	9
$P_{60} \leq \text{SNR} < P_{80}$	14
$P_{80} \leq \text{SNR} \leq P_{100}$	17

Em seguida, realiza-se a simulação aplicando os valores de SNR de cada *xApp* à função de mapeamento, de modo a gerar um vetor de índices de MCS correspondentes. O vetor resultante do *xApp* de 1 ms é utilizado como referência para as decisões esperadas de MCS durante a medição, permitindo assim avaliar a acurácia dos *xApps* adaptativo e estático com 2 ms por meio da comparação direta com essa referência. A Figura 4.7 ilustra esse processo.

³Sem perda de generalidade, essa composição de MCSs foi escolhida para simplificar a análise e o número de experimentos de validação da proposta desta dissertação. É possível obter o mesmo resultado qualitativo considerando uma tabela com mais MCSs, pois mais níveis de SNR dificultarão as ações do algoritmo adaptativo proposto, mas também aumentarão o erro do *xApp* de 2 ms, utilizado para comparação de desempenho.

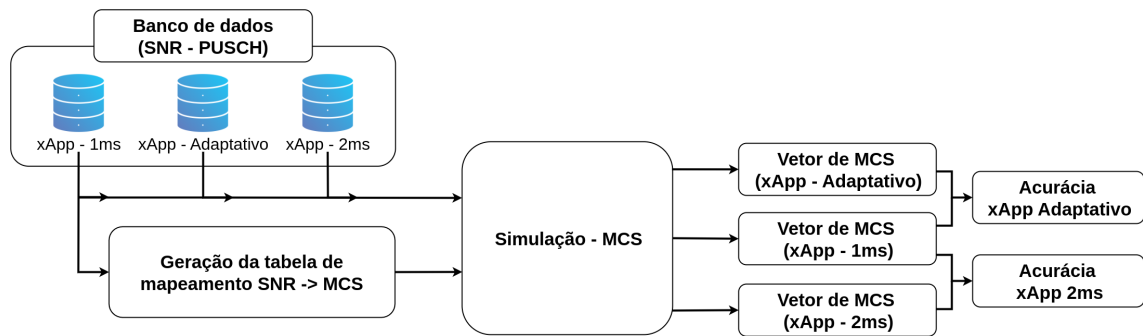


Figura 4.7: Diagrama para definição de acurácia.

O próximo capítulo apresenta e discute os resultados de prova de conceito desta dissertação.

Capítulo 5

Resultados

Neste capítulo, são apresentados e discutidos os resultados obtidos a partir da aplicação da metodologia descrita no Capítulo 4, considerando os dois procedimentos de medição adotados e o algoritmo proposto no Capítulo 3.

5.1 Definição do Limiar de Decisão

Como definido na Seção 4.2, inicialmente é necessário realizar a **Definição do Limiar de Decisão**. Essa etapa começa com a calibração do algoritmo de decisão do *xApp* adaptativo, ou seja, definir o valor do limiar θ para a métrica SNR do PUSCH, no cenário em que será aplicado. Conforme a Figura 4.5, a definição de θ baseia-se na CDF dos valores do vetor $\mathbf{v}\sigma_{\Delta}$. Para a obtenção desses dados, no cenário ilustrado na Figura 4.4, foi conduzida uma campanha de medição com duração total de 10 horas, abrangendo diferentes níveis de ocupação do laboratório, incluindo períodos de alto, médio e baixo fluxo de pessoas. A Figura 5.1 apresenta o histograma e a respectiva CDF dos valores de $\mathbf{v}\sigma_{\Delta}$, permitindo observar a distribuição do conjunto de dados. Nota-se que o percentil de 50% ocorre em $\sigma_{\Delta} = 1,51$, sendo este o valor adotado como limiar θ do algoritmo adaptativo.

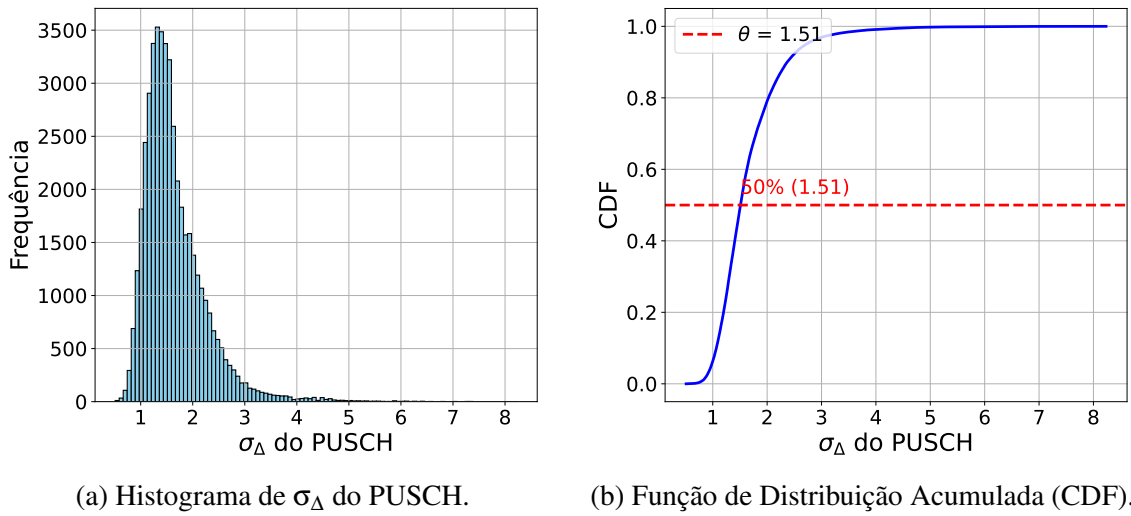


Figura 5.1: Distribuição estatística dos valores de σ_{Δ} para definição de θ .

Antes de prosseguir para os resultados da **Avaliação de Desempenho**, foi realizado um teste preliminar com o objetivo de validar o funcionamento do *xApp* adaptativo utilizando o limiar $\theta = 1,51$. Esse teste foi conduzido com a métrica SNR do canal PUSCH, o KPI monitorado pelo *xApp*. A Figura 5.2a apresenta a variação dos valores de σ_{Δ} ao longo de 20 *Duty-cycles*, permitindo observar sua flutuação em torno do limiar θ , representado pela linha tracejada em vermelho. Em complemento, a Figura 5.2 ilustra a definição dinâmica do tempo de amostragem durante esses mesmos 20 *Duty-cycles*, evidenciando o funcionamento esperado do algoritmo adaptativo. Nota-se que, em períodos com menor variação (valores de σ_{Δ} abaixo do limiar), classificados como situações de baixo risco, o tempo de amostragem é ajustado para 2 ms (frequência reduzida), promovendo a economia de recursos computacionais.

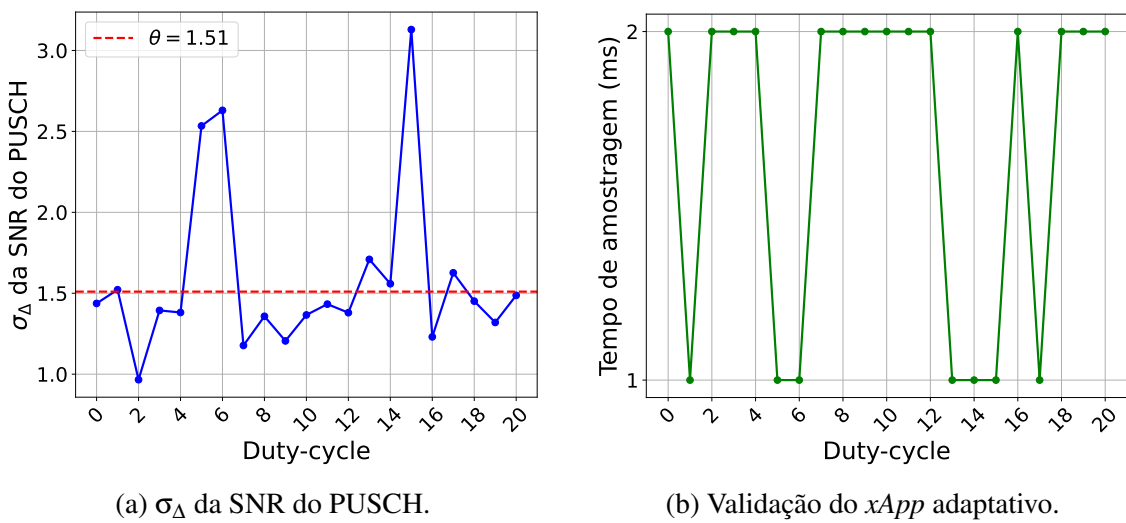


Figura 5.2: Funcionamento adaptativo: σ_{Δ} e troca de frequência de amostragem.

Com o intuito de analisar o comportamento do *xApp* adaptativo no domínio do tempo, foi elaborado o gráfico apresentado na Figura 5.3, o qual representa a variação da métrica SNR do canal PUSCH ao longo de 10 minutos. Esse gráfico é composto pelas amostras coletadas simultaneamente pelos três *xApps* avaliados: adaptativo, estático com tempo de amostragem de 1 ms e estático com 2 ms. Essa visualização tem como objetivo fornecer uma perspectiva temporal do mecanismo de adaptação ilustrado na Figura 5.2, permitindo observar o impacto do algoritmo adaptativo no processo de aquisição de amostras. Adicionalmente, são apresentados a seguir dois trechos específicos desse sinal, os quais ilustram as duas situações operacionais do *xApp* adaptativo: quando $\sigma_{\Delta} < \theta$ e quando $\sigma_{\Delta} \geq \theta$.

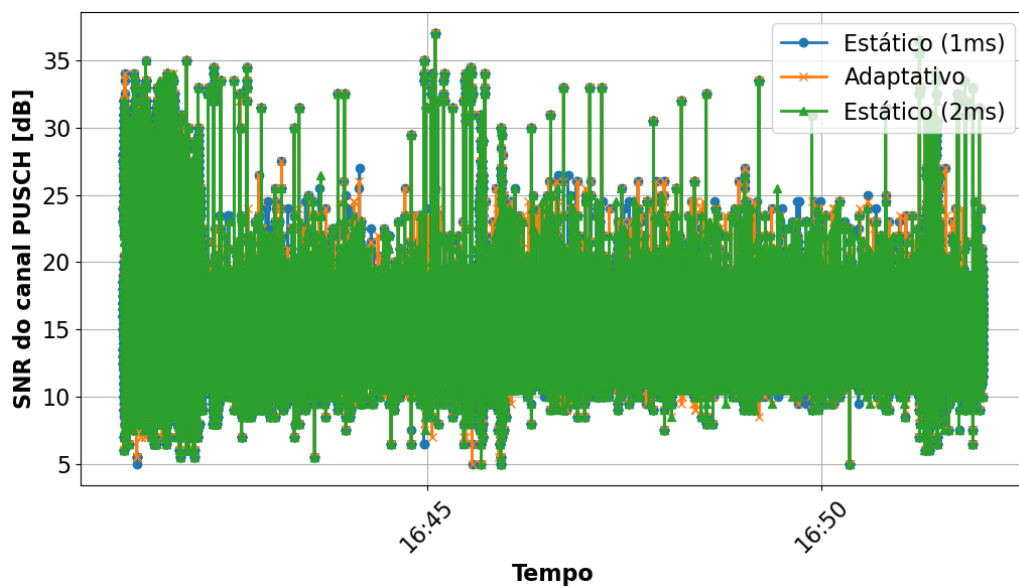


Figura 5.3: SNR do canal PUSCH em função do tempo.

A Figura 5.4 detalha um trecho do gráfico temporal em que há variações rápidas na métrica SNR do canal PUSCH. Nessa situação, o *xApp* adaptativo opta por manter o tempo de amostragem em 1 ms, interpretando o comportamento do canal como uma situação de alto risco. Nota-se que, ao adotar essa frequência de amostragem mais elevada, o *xApp* adaptativo consegue capturar com precisão as flutuações da métrica, enquanto o *xApp* estático configurado com tempo de 2 ms falha em registrar adequadamente os picos de variação. Essa comparação evidencia a vantagem do mecanismo adaptativo em contextos de instabilidade, maximizando a representatividade das amostras coletadas sem perda de informação relevante.

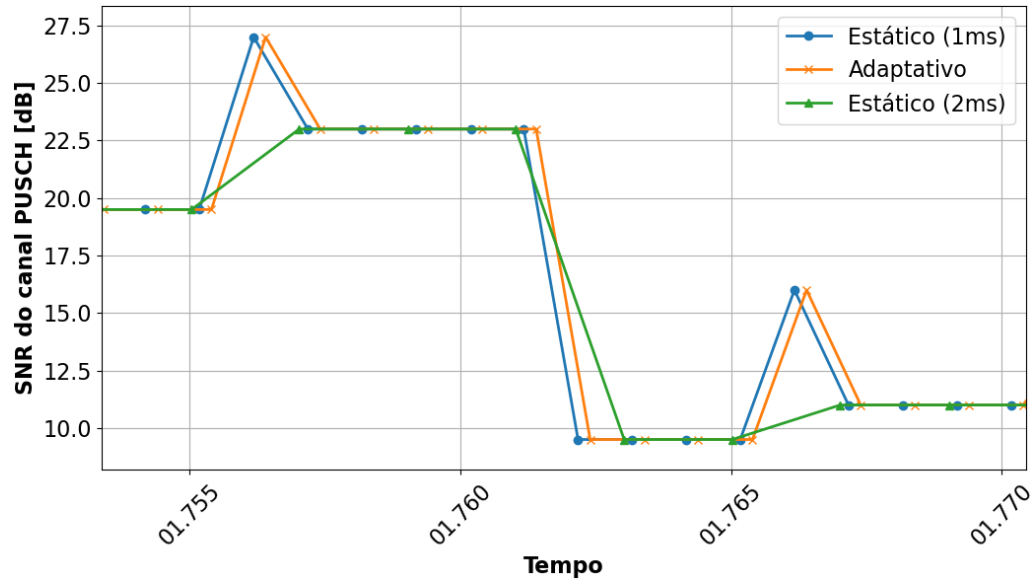


Figura 5.4: *xApp* adaptativo operando com alta frequência de amostragem.

Por outro lado, a Figura 5.5 apresenta um trecho em que a métrica SNR apresenta comportamento mais estável, com variações suaves ao longo do tempo. Nessa condição, o *xApp* adaptativo reconheceu a baixa variabilidade do sinal e, como resultado, aumentou o tempo de amostragem para 2 ms (menor frequência de coleta), visando à economia de recursos computacionais. Vale ressaltar que, mesmo com essa redução na frequência de amostragem, o *xApp* adaptativo continua obtendo amostras representativas e sem perda significativa de informação, conforme observado pela similaridade com o sinal amostrado pelo *xApp* de referência (tempo estático em 1 ms). Esse comportamento evidencia a capacidade do algoritmo adaptativo de economizar recursos sem comprometer a coleta de informações do KPI monitorado.

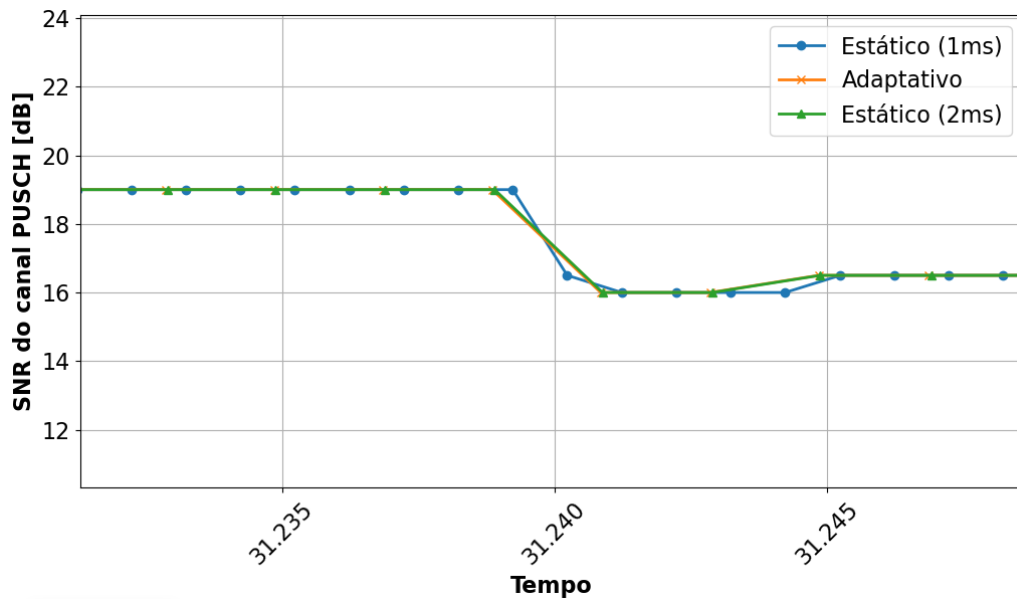


Figura 5.5: *xApp* adaptativo operando com frequência de amostragem reduzida.

5.2 Avaliação de Desempenho

A avaliação de desempenho começa com a análise do consumo de CPU e da influência dos parâmetros DC e α sobre esse consumo. Considerando o cenário descrito na Seção 4.1 e aplicando a metodologia detalhada na Seção 4.2, foi conduzida uma bateria de medições, com duração de 20 minutos para cada combinação de DC e α . Essa etapa tem por objetivo construir, com valores reais, o gráfico baseado no grid apresentado anteriormente na Figura 4.6.

A Figura 5.6 apresenta o gráfico resultante das medições, representando a economia percentual de CPU, com intervalo de confiança de 95%, obtida pelo *xApp* adaptativo em relação ao *xApp* estático de 1 ms, para cada combinação de DC e α . Pode-se observar que os maiores ganhos de economia de CPU ocorrem em configurações com DC elevado e α reduzido, com destaque para a combinação ($DC=1,0$ e $\alpha = 0,1$), que alcançou uma economia de 33,8%. À medida que α se aproxima de DC , a economia tende a diminuir, chegando a valores negativos nas configurações ($DC = 0,4$ e $\alpha = 0,3$) e ($DC = 0,6$ e $\alpha = 0,5$). Isso ocorre porque a fase KEP passa a ocupar praticamente todo o *Duty-cycle*, elevando significativamente o volume de dados processados na avaliação de risco, por consequência, aumentando o consumo de CPU. Esses resultados evidenciam que a escolha adequada dos parâmetros é crucial para maximizar a eficiência computacional do *xApp* adaptativo.

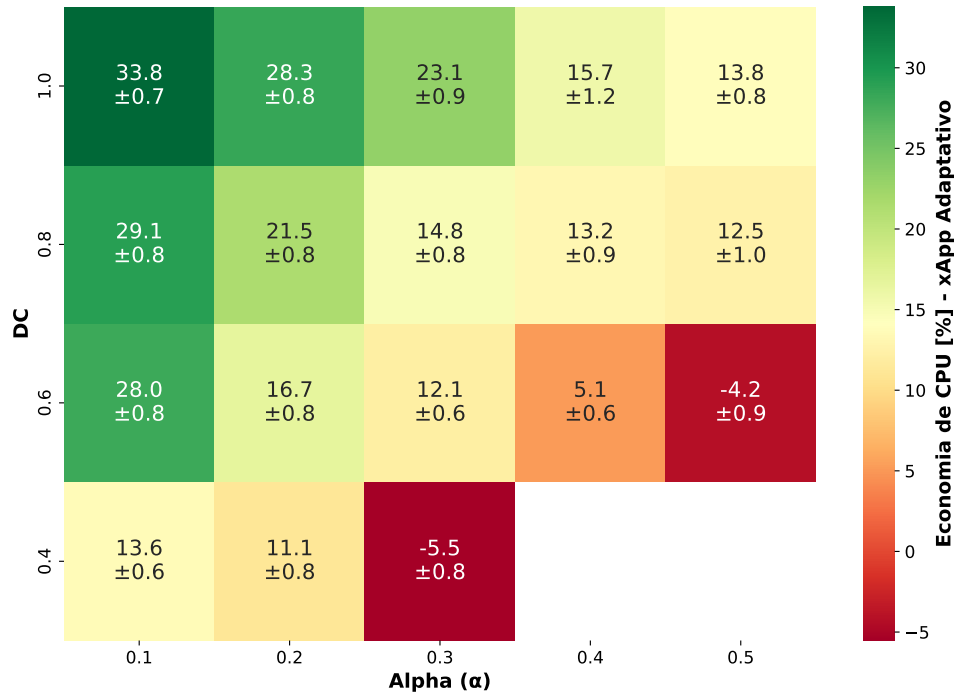


Figura 5.6: Grid de economia de CPU - *xApp* Adaptativo vs Estático de 1ms.

Além das medições que resultaram no grid do *xApp* adaptativo, também foi executado, de forma simultânea, o *xApp* estático com amostragem de 2 ms. Esse *xApp* representa uma abordagem tradicional para a redução de consumo computacional, consistindo apenas na configuração de um maior tempo fixo entre a coleta de amostras. Com essa abordagem, conseguiu-se uma economia de CPU de **42,9%** em relação ao *xApp* estático de 1 ms. Como esperado, essa economia supera o melhor caso do *xApp* adaptativo, o qual conseguiu uma economia de **33,8%** com a configuração ($DC = 1,0$ e $\alpha = 0,1$). Contudo, essa economia é alcançada ao custo de uma menor fidelidade na representação do KPI, o que impacta diretamente o desempenho em aplicações práticas. A intenção do *xApp* adaptativo é disponibilizar uma solução de compromisso entre economia de CPU e perda de precisão na coleta de KPIs.

Para quantificar o impacto do *xApp* adaptativo na perda de precisão na coleta de KPIs, a análise de desempenho estende-se ao compromisso entre economia de CPU e a acurácia. Conforme a metodologia detalhada na Seção 4.2, a acurácia é avaliada por meio da tomada de decisão do MCS com base na métrica de SNR do PUSCH. Nesse processo, ilustrado na Figura 4.7, a sequência de decisões de MCS gerada a partir das amostras do *xApp* de 1 ms é considerada como o gabarito de referência. Assim, a acurácia do *xApp* adaptativo e do *xApp* estático de 2 ms é o percentual de vezes que suas decisões de MCS coincidiram com a referência.

Nesse contexto, para a análise de compromisso, foram escolhidas quatro configurações de DC e α do grid de economia de CPU da Figura 5.6. As configurações escolhidas e o modo de economia correspondente a cada uma delas são:

- ($DC = 1,0$ e $\alpha = 0,1$): *xApp* adaptativo (**Economia máxima**)

- ($DC = 0,8$ e $\alpha = 0,2$): *xApp* adaptativo (**Economia média**)
- ($DC = 0,4$ e $\alpha = 0,2$): *xApp* adaptativo (**Economia baixa**)
- ($DC = 0,6$ e $\alpha = 0,4$): *xApp* adaptativo (**Economia mínima**)

A Tabela 5.1 consolida essa análise, comparando o desempenho do *xApp* estático de 2 ms com as quatro configurações escolhidas. Os resultados permitem correlacionar diretamente a economia de recursos computacionais com a capacidade de cada *xApp* realizar decisões de MCS corretas.

Tabela 5.1: Compromisso entre economia de CPU e acurácia na decisão de MCS.

Cenário	Configuração (DC, α)	Economia de CPU [%]	Acurácia de MCS [%]
Estático 2 ms	N/A	42,90	77,30
<i>Adaptativo (Economia Máxima)</i>	(1,0 e 0,1)	33,80	79,80
<i>Adaptativo (Economia Média)</i>	(0,8 e 0,2)	21,48	81,25
<i>Adaptativo (Economia Baixa)</i>	(0,4 e 0,2)	11,12	83,73
<i>Adaptativo (Economia Mínima)</i>	(0,6 e 0,4)	5,14	86,01

Ao analisar os resultados da Tabela 5.1, observa-se que o *xApp* estático de 2 ms, apesar de apresentar a maior economia de CPU (**42,9%**), também registra a menor acurácia na seleção de MCS (**77,3%**). Comparando-o com as diferentes configurações do *xApp* adaptativo, percebe-se que é necessário um considerável gasto de CPU para aumentar a acurácia de MCS. Em relação à configuração com *Economia Máxima*, é preciso reduzir a economia de CPU de 42,9% para 33,8% para se conseguir um aumento de 2,5 pontos percentuais na acurácia de MCS. No outro extremo, com *Economia Mínima*, a economia de CPU se reduz a 5,14% para um aumento de 8,71 pontos percentuais na acurácia. Em um primeiro momento, pode-se pensar que o compromisso do consumo de CPU para o ganho de acurácia obtido não seja compensador. No entanto, é importante considerar a mudança de paradigma possibilitada pelo *xApp* adaptativo e os benefícios decorrentes de sua flexibilidade.

No contexto do modelo de monitoramento tradicional e estático, o operador limita-se a uma escolha fixa do tempo de amostragem, que não se adapta às flutuações do ambiente de rádio. Nesse sentido, para uma determinada aplicação não crítica que suportasse uma acurácia mínima de 80%, o operador não poderia optar pela economia de CPU utilizando o *xApp* estático de 2ms, que possui apenas 77,3% de acurácia. Por outro lado, poderia optar pelo *xApp* adaptativo em sua configuração de *Economia Média*, atingindo o requisito mínimo da aplicação e fornecendo uma economia de 21,48% de CPU, que não seria possível com os *xApps* estáticos.

Ao considerar o contexto em que o monitoramento está inserido, essa economia pode ser de alto impacto. Conforme discutido na Seção 2.2, tanto o tráfego na interface E2 quanto a quantidade de KPIs reportada por nó E2 evidenciam desafios significativos de eficiência energética em implantações de larga escala. Nesse cenário, mesmo economias

aparentemente modestas, como 21,48%, podem resultar em ganhos expressivos no custo computacional total.

Para quantificar essa economia em termos energéticos, estimou-se a potência associada a um núcleo de CPU dedicado, com base no *datasheet* do processador utilizado [36], resultando em aproximadamente 15,31 W. Considerando esse valor, o tempo de execução e o uso médio de CPU, o consumo energético do *xApp* adaptativo durante os 20 minutos de medição foi de 1.853,73 J, enquanto o do *xApp* estático de referência atingiu 2.360,84 J.

Projetando-se esse consumo para um monitoramento de apenas 1 dia, no cenário de larga escala proposto em [14], composto por 300 nós E2 e 100 KPIs por nó, a economia proporcionada pelo *xApp* adaptativo alcançaria 1.095.265.152 J, o equivalente a 304,24 kWh. O mesmo estudo aponta que a quantidade de KPIs por nó pode crescer consideravelmente, especialmente em contextos nos quais a coleta de métricas por UE é essencial. Nesse sentido, considerando 200 KPIs por nó E2, a economia fornecida pelo *xApp* adaptativo chegaria a 608,48 kWh por dia, totalizando 18.254,4 kWh ao longo de um mês. Do ponto de vista ambiental, segundo calculadora disponibilizada pela ONG SOS Mata Atlântica [37], essa economia corresponde a uma redução mensal de aproximadamente 0,99 tonelada na emissão de CO_2 .

Capítulo 6

Conclusão

Este trabalho propôs e avaliou um *xApp* adaptativo para monitoramento de métricas em redes Open RAN 5G, com o objetivo de proporcionar a flexibilidade de explorar o compromisso do uso de recursos computacionais sem comprometer a eficácia do monitoramento de KPIs da interface E2. Inicialmente, identificou-se que a coleta contínua de KPIs em alta frequência resulta em elevado consumo de CPU e energia, o que pode impactar negativamente a eficiência do sistema. Diante desse cenário, a solução desenvolvida ajusta dinamicamente a frequência de coleta dos KPIs com base na variabilidade observada nos dados, permitindo uma atuação responsiva às condições da rede. Além disso, o *xApp* propõe um novo paradigma de monitoramento, oferecendo ao operador a flexibilidade de configurar o algoritmo em diferentes níveis de economia, de acordo com os requisitos específicos de cada aplicação.

Os resultados demonstraram que, embora o monitoramento estático de baixa frequência (2ms) tenha apresentado a maior economia de CPU, sua aplicabilidade é restrita a cenários cujos requisitos sejam compatíveis com sua acurácia, que se mostrou a mais baixa entre todas as abordagens avaliadas. Por outro lado, a abordagem adaptativa, ainda que apresente uma economia de CPU inferior, destaca-se por sua flexibilidade de configuração, permitindo ajustar o nível de economia de recursos conforme as exigências específicas da aplicação. Essa adaptabilidade viabiliza seu uso em contextos nos quais o *xApp* estático de baixa frequência não atende aos requisitos de desempenho. Além disso, a economia fornecida pela solução adaptativa se torna especialmente relevante em cenários de larga escala, alcançando economia de até 608,48 kWh por dia de monitoramento, o que evidencia seu potencial para reduzir o impacto energético do sistema.

Por fim, como continuidade deste trabalho, sugere-se: (i) a investigação de técnicas de inteligência artificial para o aprimoramento do algoritmo de decisão, superando o modelo estatístico baseado em desvio padrão, com o objetivo de aumentar a acurácia do *xApp* adaptativo; (ii) a avaliação do desempenho de *xApps* de controle (como alocação de recursos), quando integrados ao monitoramento adaptativo em vez do estático; (iii) o desenvolvimento de uma interface de configuração que permita o ajuste dinâmico dos parâmetros do *xApp* (DC e α), com base no perfil de tráfego, tipo de serviço ou categoria de usuário; e (iv) a ampliação do escopo de monitoramento para outras métricas, bem como a avaliação da solução em cenários com diferentes perfis de mobilidade.

6.1 Produção Acadêmica

Durante o mestrado, o autor participou no desenvolvimento de 6 trabalhos junto ao grupo de pesquisa GppCom, com atuação no centro de pesquisa LANCE. Os trabalhos são:

- Artigos de congresso:
 - Monitoramento Unificado para Multiplataformas E2 em Sistemas Open RAN. XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2025 (Aceito para apresentação);
 - Qualidade de Vídeo em Redes de Acesso Fixo sem Fio com Gateways 5G: Uma Avaliação Objetiva. XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2025 (Aceito para apresentação);
 - Metodologia para Geração de Radio Environment Maps de Redes 5G SA Privadas. XLII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2024;
 - Licenciamento de Espectro para uso Científico: Legislação e Estudo de Caso de Sistema 5G Aberto. XIV Conferência Nacional em Comunicações, Redes e Segurança da Informação (ENCOM), 2024 (prêmio de melhor artigo).
- Artigo de Jornal:
 - *Virtualized 5G Testbed using OpenAirInterface: Tutorial and Benchmarking Tests. Journal of Internet Services and Applications*, Vol. 15 No. 1 (2024).
- Patente:
 - Ideia de patente aceita internamente em projeto em parceria com a Lenovo e pedido de patente nos Estados Unidos da América com o título *Adaptive Monitoring of Radio Intelligent Controller Key Performance Indicators*.

Dentre os trabalhos desenvolvidos, destaca-se o pedido de patente que fornece a ideia geral para a aplicação específica desenvolvida neste trabalho de mestrado.

Referências Bibliográficas

- [1] RESEARCH, P. *Open RAN Market Size to Hit USD 38.71 Billion by 2034*. 2025. <https://www.precedenceresearch.com/open-ran-market>.
- [2] O-RAN Alliance. *O-RAN-WG1: O-RAN Architecture Description - v12.00*. [S.l.]: Tech. Spec., 2024.
- [3] WANI, M. et al. Open ran: A concise overview. *IEEE Open Journal of the Communications Society*, v. 6, p. 13–28, 2025.
- [4] SINGH, S.; SAMAL, U. Insights and trends in open ran: The future of mobile networks. *Journal of Network and Systems Management*, Springer, v. 33, n. 1, p. 46, 2025.
- [5] ITU-R. Report ITU-R M.2320 - Future technology trends of terrestrial IMT systems. Disponível em: www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2320-2014-MSW-E.docx.
- [6] HOLMA, H.; TOSKALA, A.; NAKAMURA, T. (Ed.). *5G Technology: 3GPP Evolution to 5G-Advanced*. 2. ed. Chichester: John Wiley & Sons, 2024. ISBN 978-1-119-81603-4.
- [7] MARTIAN, A. et al. Towards open ran in beyond 5g networks: Evolution, architectures, deployments, spectrum, prototypes, and performance assessment. *Computer Networks*, v. 259, p. 111087, 2025. ISSN 1389-1286. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128625000556>.
- [8] AZARIAH, W. et al. A survey on open radio access networks: Challenges, research directions, and open source approaches. *Sensors*, v. 24, p. 1038, 2024. Disponível em: <https://doi.org/10.3390/s24031038>.
- [9] WANG, T. H. et al. Design of a network management system for 5g open ran. In: *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. [s.n.], 2021. p. 138–141. Disponível em: <https://doi.org/10.23919/APNOMS52696.2021.9562627>.
- [10] ALLIANCE, O.-R. *About O-RAN ALLIANCE*. Disponível em: <https://www.o-ran.org/about>.
- [11] OpenRAN@Brasil. *OpenRAN@Brasil: site oficial do programa OpenRAN@Brasil*. 2025. Disponível em: <https://openranbrasil.org.br/>.

- [12] Ministério da Ciência, Tecnologia e Inovação (MCTI) & Embrapii. *MCTI e Embrapii anunciam R\$ 180 milhões para Centros de Competência em tecnologia (Open RAN, 5G/6G e mundos virtuais)*. 2023. Disponível em: <<https://shorturl.at/Fvi5U>>.
- [13] YOO, H. M. et al. Load balancing algorithm running on open ran ric. In: *2022 International Conference on ICT Convergence (ICTC)*. [s.n.], 2022. p. 1226–1228. Disponível em: <<https://doi.org/10.1109/ICTC55196.2022.9952635>>.
- [14] LIMA, J. P. S. H.; KATSAROS, G. N.; NIKITOPOULOS, K. Power-efficient ran intelligent controllers through optimized kpi monitoring. In: *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. [S.l.: s.n.], 2025. p. 1–6.
- [15] LIANG, X. et al. Enhancing energy efficiency in o-ran through intelligent xapps deployment. In: *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. [S.l.: s.n.], 2024. p. 1–6.
- [16] GNANASEKARAN, J. K.; BOUTIBA, K.; KSENTINI, A. Greening 5g : Empowering dynamic drx with deep reinforcement learning and o-ran. In: *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. [S.l.: s.n.], 2025. p. 1–2.
- [17] HABIB, M. A. et al. Transformer-based wireless traffic prediction and network optimization in o-ran. In: *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. [S.l.: s.n.], 2024. p. 1–6.
- [18] HOFFMANN, M.; KRYSZKIEWICZ, P. O-ran for energy-efficient serving cluster formulation in user-centric cell-free mmimo. In: *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. [S.l.: s.n.], 2024. p. 1–2.
- [19] MUNGARI, F. et al. Oreo: O-ran intelligence orchestration of xapp-based network services. In: *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications*. [S.l.: s.n.], 2024. p. 71–80.
- [20] HABIB, M. A. et al. Intent-driven intelligent control and orchestration in o-ran via hierarchical reinforcement learning. In: *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. [S.l.: s.n.], 2023. p. 55–61.
- [21] POLESE, M. et al. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE COMMUNICATIONS SURVEYS AND TUTORIALS*, v. 25, n. 2, p. 1376–1411, 2023.
- [22] AZARIAH, W. et al. A Survey on Open Radio Access Networks: Challenges, Research Directions, and Open Source Approaches. *Sensors*, v. 24, n. 3, p. 1–49, 2024. ISSN 14248220.

- [23] MARINOVA, S.; LEON-GARCIA, A. Intelligent o-ran beyond 5g: Architecture, use cases, challenges, and opportunities. *IEEE Access*, v. 12, p. 27088–27114, 2024.
- [24] COLETTI, C. et al. *O-RAN: Towards an Open and Smart RAN*. 2018. White paper. Disponível em: <<https://www.o-ran.org/resources>>.
- [25] ALAVIRAD, M. et al. O-RAN architecture, interfaces, and standardization: Study and application to user intelligent admission control. *Frontiers in Communications and Networks*, v. 4, n. March, p. 1–17, 2023. ISSN 2673530X.
- [26] GARCIA-SAAVEDRA, A.; COSTA-PEREZ, X. O-RAN: Disrupting the Virtualized RAN Ecosystem. *IEEE Communications Standards Magazine*, v. 5, n. 4, p. 96–103, 2021. ISSN 24712825.
- [27] SANTOS, J. F. et al. Managing o-ran networks: xapp development from zero to hero. *IEEE Communications Surveys & Tutorials*, p. 1–1, 2025.
- [28] 3RD Generation Partnership Project; Telecommunication management; Management and orchestration; 5G performance measurements (Release 18). [S.l.], 2024. Accessed: 2025-01-06. Disponível em: <<https://www.3gpp.org/ftp/Specs/html-info/28552.htm>>.
- [29] KALTENBERGER, F. et al. Driving innovation in 6g wireless technologies: The openairinterface approach. *Computer Networks*, v. 269, p. 111410, 2025. ISSN 1389-1286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1389128625003779>>.
- [30] KALTENBERGER, F. et al. OpenAirInterface: Democratizing innovation in the 5G Era. *Computer Networks*, Elsevier B.V., v. 176, n. May, p. 107284, 2020. ISSN 13891286. Disponível em: <<https://doi.org/10.1016/j.comnet.2020.107284>>.
- [31] SCHMIDT, R.; IRAZABAL, M.; NIKAEIN, N. FlexRIC: An SDK for next-generation SD-RANs. *CoNEXT 2021 - Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, p. 411–425, 2021.
- [32] OAI. *Core Network 5G, v1.5.0*. 2024. https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed/-/tree/v1.5.0?ref_type=tags. Acessado em: julho de 2025.
- [33] OAI. *RAN, 2023.w37*. 2024. https://gitlab.eurecom.fr/oai/openairinterface5g/-/tree/2023.w37?ref_type=tags. Acessado em: julho de 2025.
- [34] CAMPOS, A. et al. Licenciamento de espectro para uso científico: legislação e estudo de caso de sistema 5g aberto. In: *Anais da Conferência Nacional em Comunicações, Redes e Segurança da Informação (ENCOM)*. Natal, RN: IFRN, 2024. Disponível em: <<https://www.even3.com.br/anais/encom2024/889809>>.

- [35] Zabbix SIA. *Zabbix Features*. 2025. <https://www.zabbix.com/documentation/current/en/manual/introduction/features>. Acessado em: julho de 2025.
- [36] Intel Corporation. *Intel® Core™ i7-12700 Processor (25M Cache, up to 4.90GHz) – Specifications*. [S.l.]: Intel Corporation, 2022. <https://www.intel.com/content/www/us/en/products/sku/134591/intel-core-i7-12700-processor-25m-cache-up-to-4-90-ghz/specifications.html>. Acessado em julho de 2025.
- [37] SOS Mata Atlântica. *Calcule sua emissão de CO₂*. 2025. <https://www.sosma.org.br/calcule-sua-emissao-de-co2>. Acessado em julho de 2025.