

Análise de Desempenho do Overhead Computacional de xApps Baseadas em Aprendizado por Reforço em Open-RAN

Robert V. O. Gonçalves¹, Elvis M. G. Stancanelli¹, Emanuel F. Coutinho¹,
Francisco Helder C. dos S. Filho¹

¹ Programa de Pós Graduação em Computação – Universidade Federal do Ceará
Caixa Postal 63902.580 – Quixadá – CE – Brasil

robertvinicius@alu.ufc.br,
{elvis.stancanelli, emanuel.coutinho, helderhdw}@ufc.br

Abstract. *The Open Radio Access Network (Open-RAN) architecture enables intelligent applications (xApps) to run within the Near-Real-Time RAN Intelligent Controller (Near-RT RIC), thereby allowing the application of Deep Reinforcement Learning (DRL) techniques for automated RAN control. This paper evaluates the computational overhead of DRL-based xApps by comparing the performance of Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) algorithms against a baseline policy. The simulations were conducted in an Open-RAN environment based on the Nori Blueprint v1, considering eMBB and URLLC service profiles and different levels of artificial computational workload. Results indicate that decision-making latency increases consistently with the applied computational load. Although DRL-based algorithms exhibit a higher overhead, their execution times remain strictly compliant with the temporal constraints of the Near-RT RIC.*

Resumo. *A arquitetura Open Radio Access Network (Open-RAN) viabiliza a execução de aplicações inteligentes (xApps) no Near-Real-Time RAN Intelligent Controller (Near-RT RIC), propiciando a integração de técnicas de Aprendizado por Reforço Profundo (DRL) para a automação do controle da RAN. Este trabalho apresenta uma análise de desempenho do overhead computacional de xApps baseadas em DRL, comparando os algoritmos Deep Q-Network (DQN) e Proximal Policy Optimization (PPO) com uma política baseline. As simulações foram realizadas em um ambiente Open-RAN baseado no Blueprint Nori v1, considerando perfis de serviço eMBB e URLLC e diferentes níveis de carga computacional artificial. Os resultados indicam aumento consistente do tempo de decisão com a elevação da carga computacional avaliada, sendo mais elevado para algoritmos baseados em DRL, mas permanecendo compatível com os requisitos temporais do Near-RT RIC.*

1. Introdução

Nas últimas décadas, a arquitetura das Redes de Acesso por Rádio (RAN) tem passado por transformações significativas, impulsionadas pelo aumento do tráfego móvel, pela densificação das redes e pela necessidade de soluções mais flexíveis e escaláveis. As redes de nova geração demandam paradigmas capazes de reconfigurar suas operações em tempo real, de modo a se adaptar às variações de tráfego e atender requisitos heterogêneos de Qualidade de Serviço (QoS) [Tsampazi et al. 2023]. Nesse contexto, uma das abordagens

que ganhou destaque na evolução das redes 5G e nos estudos para o 6G é o *Open Radio Access Network* (Open-RAN) [Singh et al. 2020].

O Open-RAN representa uma mudança importante em relação ao modelo tradicional de RAN, que, por ser monolítico e de código fechado, limita a interoperabilidade e a customização de serviços [Menezes et al. 2023]. Ao promover a desagregação entre a Radio Unit (O-RU), a Distributed Unit (O-DU) e a Central Unit (O-CU), o Open-RAN substitui interfaces proprietárias por interfaces abertas e componentes interoperáveis de múltiplos fornecedores [Menezes et al. 2023]. Essa arquitetura, fortemente apoiada em virtualização e componentes baseados em software, tem se mostrado uma alternativa promissora para ampliar a inovação e reduzir a dependência de fabricantes específicos.

O pilar central para viabilizar a alocação inteligente no Open-RAN é o Controlador Inteligente de RAN (RIC). O RIC desacopla as funções de controle da RAN, permitindo que a rede seja programável e otimizada a partir de dados [Polese et al. 2023]. Ele é dividido em dois componentes principais: o Non-Real-Time (Non-RT) RIC, responsável por funções de controle em escalas de tempo mais longas, como o treinamento de modelos de IA/ML, e o Near-Real-Time (Near-RT) RIC. Este último opera em laços de controle rápidos, entre 10 ms e 1 s, e hospeda aplicações de software de terceiros, denominadas xApps [Tsampazi et al. 2023]. xApps utilizam a interface E2 para coletar medições da rede e aplicar decisões de controle em ciclo fechado [Menezes et al. 2023, Polese et al. 2023].

Esse paradigma de controle em ciclo fechado, orientado por dados, é compatível com técnicas de Aprendizado por Reforço Profundo (DRL), que dispensam um modelo explícito da dinâmica da rede [Tsampazi et al. 2023]. O DRL tem sido amplamente investigado como ferramenta para a automação do Open-RAN, permitindo que agentes aprendam políticas de alocação de recursos para lidar com objetivos complexos e, muitas vezes, conflitantes. Trabalhos recentes demonstram a eficácia de xApps baseadas em DRL para otimização de slicing e gerenciamento de trade-offs de QoS [Polese et al. 2023]. No entanto, o design desses agentes, especialmente no que se refere à função de recompensa e à complexidade do processo de decisão, impacta diretamente o desempenho e a viabilidade prática dessas soluções.

Nesse contexto, este artigo apresenta uma análise de desempenho da alocação inteligente de recursos em Open-RAN, com foco no custo computacional e na viabilidade temporal de xApps baseadas em DRL. O objetivo é comparar os algoritmos Proximal Policy Optimization (PPO), representante da abordagem *on-policy*, e Deep Q-Network (DQN), representante da abordagem *off-policy*, analisando como o aumento da carga computacional influencia o tempo de decisão de xApp (overhead) e sua adequação aos requisitos temporais associados a diferentes perfis de serviço, como eMBB e URLLC. Neste estudo, o *overhead* computacional é caracterizado como a latência de processamento intrínseca à execução do algoritmo de decisão de xApp, medida em milissegundos por ciclo a cada iteração do laço de controle no Near-RT RIC, sendo utilizada como indicador de viabilidade temporal. No Aprendizado por Reforço, algoritmos *on-policy* atualizam sua política com base nas experiências geradas pela própria política corrente, enquanto algoritmos *off-policy* podem aprender a partir de experiências coletadas por políticas distintas. O PPO representa a abordagem *on-policy*, enquanto o DQN representa a abordagem *off-policy*.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta os fundamentos teóricos; a Seção 3 discute os trabalhos relacionados; a Seção 4 descreve a metodologia adotada; a Seção 5 apresenta os resultados obtidos; e, por fim, a Seção 6 traz as conclusões e perspectivas para trabalhos futuros.

2. Fundamentação Teórica

Esta seção apresenta os conceitos fundamentais da arquitetura Open-RAN, descreve como o Aprendizado por Reforço Profundo (DRL) se aplica ao gerenciamento de recursos e discute os algoritmos PPO e DQN, que fundamentam a análise de desempenho computacional e temporal realizada neste trabalho.

2.1. A Arquitetura Open-RAN e seus Controladores Inteligentes

A arquitetura Open-RAN estabelece a desagregação da rede de acesso por rádio em elementos modulares (O-RU, O-DU e O-CU) e fomenta interfaces abertas, o que potencializa a interoperabilidade e independência de fornecedores, impulsionando inovação e automação (O-RAN Alliance, 2025). O uso de controladores inteligentes, conhecidos como RAN Intelligent Controllers (RICs), é um diferencial chave desse modelo, ao permitir a integração de técnicas baseadas em aprendizado de máquina e políticas adaptativas para otimização da performance da rede [Alam et al. 2026, Soltani et al. 2025].

O Non-Real-Time RIC (Non-RT RIC) é responsável pela orquestração de políticas de rede, análise de dados históricos e treinamento de modelos de aprendizado de máquina em escalas de tempo mais longas do que 1 segundo, integrando funcionalidades de planejamento e auto-organização por meio da interface A1 [O-RAN Software Community (O-RAN SC) 2025]. Já o Near-Real-Time RIC (Near-RT RIC) atua entre 10 ms e 1 s, próximo à borda da rede, operacionalizando decisões inteligentes (por exemplo, escalonamento, handover ou balanceamento de carga), bem como avaliando o custo computacional e a viabilidade temporal de xApps sob diferentes condições de carga [Thieu et al. 2023].

2.2. Aprendizado por Reforço Profundo (DRL) para Alocação de Recursos

O DRL vem se consolidando como uma abordagem atrativa para a resolução de problemas complexos de alocação de recursos em redes móveis e ambientes Open-RAN. Essa técnica combina o poder de algoritmos de reforço tradicionais com redes neurais profundas, permitindo que agentes aprendam políticas otimizadas diretamente a partir da interação sequencial com o ambiente, mesmo sem conhecimento prévio detalhado das regras do sistema [Mnih et al. 2015].

Diferentes estudos mostram que o uso de DRL supera métodos heurísticos tradicionais tanto em eficiência espectral quanto em adaptação a dinâmicas reais da rede. Por exemplo, [Bansbach et al. 2021] demonstraram que agentes DRL, ao considerar informações do buffer e demandas dos usuários, conseguem realizar escalonamento de recursos em sistemas baseados no OFDMA de modo mais eficiente e justo, mesmo diante de requisitos variados de taxa e latência.

No contexto de slicing de redes, técnicas como Double Deep Q-Network (DDQN), uma variante do DQN, têm sido empregadas para otimizar simultaneamente a potência

de transmissão e a alocação de blocos de recursos, atendendo serviços com diferentes exigências como URLLC e eMBB. Os resultados mostram desempenho superior quanto ao uso eficiente de potência e recursos se comparado a políticas convencionais [Chen et al. 2023].

Embora diversos trabalhos demonstrem ganhos diretos de KPIs de rede com o uso de DRL, este estudo concentra-se em uma perspectiva complementar: a análise do custo computacional e do tempo de decisão de agentes DRL executados como xApps no Near-RT RIC. Essa abordagem permite avaliar a viabilidade prática desses algoritmos frente às restrições temporais impostas por diferentes classes de serviço.

2.3. Proximal Policy Optimization (PPO) - On-Policy

O algoritmo PPO é amplamente utilizado em problemas de controle contínuo e oferece boa estabilidade na aprendizagem, sendo adotado em projetos experimentais de alocação dinâmica de espectro e recursos em cenários massivos de MIMO e múltiplas células [Mnih et al. 2015].

2.4. Deep Q-Network (DQN) - Off-Policy

O DQN consiste em uma combinação de Q-learning com redes neurais profundas, sendo indicado para espaços de ação discretos. Recentemente, variantes como DDQN e D3QN têm mostrado alta eficácia em problemas de escalonamento e alocação em ambientes de rede flexível, industrial e com cenários de múltiplos serviços concorrentes [Zheng 2025]. DQN e suas variantes conseguem maximizar vazão de dados, garantir níveis de latência e adaptação a mudanças abruptas no ambiente sem fio, como interferências e mobilidade dos usuários.

3. Trabalhos Relacionados

O trabalho de [Bordin et al. 2025] foca na eficiência energética, propondo um xApp baseado em DRL que decide dinamicamente sobre a ativação e desativação de interfaces radio-frequência para poupar energia. Os autores avaliam o desempenho comparando diretamente os algoritmos PPO (on-policy) e DQN (off-policy), utilizando o simulador ns-O-RAN. Embora esse estudo valide a comparação PPO vs. DQN, seu foco é singular na otimização de energia. O presente trabalho complementa essa análise ao investigar o impacto do algoritmo DRL no custo computacional e no tempo de decisão de xApps executados no Near-RT RIC, considerando diferentes níveis de carga e classes de tráfego.

[Sohaib et al. 2025] abordam um dos desafios mais críticos do 5G: a otimização de serviços URLLC (Ultra-Reliable Low Latency Communication) dentro da arquitetura O-RAN. O artigo realiza uma análise de trade-off comparando vários algoritmos DRL, como DDPG e DDQN (uma variante do DQN), para gerenciar o equilíbrio entre confiabilidade e latência. Os autores propõem um agente avançado (TD3-TS) que alcança alta confiabilidade. O trabalho é fundamental para contextualizar os requisitos de latência, mas sua análise se concentra em algoritmos off-policy mais complexos. O presente estudo se diferencia ao focar na comparação fundamental entre PPO e DQN sob a ótica do custo computacional e da latência de decisão de xApp, analisando a viabilidade desses algoritmos no Near-RT RIC sem recorrer a arquiteturas mais complexas como o TD3.

Diferentemente das abordagens baseadas em DRL, [Hussien and Jahankhani 2025] comparam técnicas de Aprendizado de Máquina

ensemble (como Random Forest e XGBoost) com métodos meta-heurísticos (como Whale Optimization Algorithm) para alocação de recursos, decompondo o problema em tarefas de classificação e regressão. Os autores concluem que, após o treinamento, os modelos produzem decisões quase instantâneas, sendo adequados para operação em tempo real. Contudo, o estudo não utiliza DRL. Em contraste, o presente trabalho investiga PPO e DQN, analisando especificamente o impacto temporal do processo de decisão sequencial no custo computacional de xApps em ambientes Open-RAN.

[Park et al. 2024] trazem um levantamento bibliográfico sobre as tendências recentes da aplicação de DRL no Open-RAN. O artigo sumariza a arquitetura Open-RAN e como o aprendizado por reforço está sendo integrado para enfrentar desafios de gerenciamento dinâmico de recursos e tomada de decisão em tempo real. Ele cataloga o uso de DRL Federado, Multi-Agent RL (MARL) e DQN para casos de uso como slicing de rede e otimização de latência. O presente estudo fornece uma análise de desempenho detalhada de PPO e DQN, com foco no custo computacional, na latência de decisão e na escalabilidade de xApps.

4. Metodologia

Esta seção descreve o ambiente de simulação, o modelo de decisão implementado no xApp, a configuração dos algoritmos de Aprendizado por Reforço Profundo e as métricas de desempenho utilizadas para a análise comparativa.

4.1. Ambiente de Simulação

Os experimentos foram conduzidos utilizando o *Blueprint Nori v1*¹, desenvolvido pelo projeto OpenRAN Brasil, executado em uma máquina virtual com Ubuntu Desktop 20.04.6, 4 vCPUs e 8 GB de RAM, seguindo as especificações mínimas recomendadas pela documentação oficial da plataforma. Os componentes do Near-RT RIC e dos xApps foram executados em contêineres Docker, permitindo a repetição controlada dos experimentos e a coleta sistemática das métricas.

O xApp avaliado foi implementado em Python e opera em um laço periódico de decisão, no qual métricas são coletadas, o algoritmo configurado é executado e o tempo de decisão é registrado. Para avaliar a escalabilidade computacional, foi introduzida uma carga artificial controlada pelo parâmetro *work units*, possibilitando observar o impacto direto da complexidade do algoritmo sobre o tempo de decisão. Esse parâmetro representa apenas carga computacional artificial controlada, não correspondendo diretamente a métricas reais da RAN. Os perfis de serviço eMBB e URLLC são considerados apenas como referência de requisitos temporais, mantendo constantes os valores de latência e vazão. Portanto, a análise foca exclusivamente na viabilidade temporal e no overhead computacional dos algoritmos quando executados como xApps no Near-RT RIC.

4.2. Modelo de Decisão do xApp

O modelo de decisão adotado aqui é implementado diretamente no nível de xApp. O xApp opera em um laço periódico de decisão, no qual, a cada iteração, são coletadas métricas disponibilizadas pelo ambiente de simulação, o algoritmo de decisão configurado é executado e o tempo necessário para completar esse processamento é registrado.

¹<https://github.com/lasseufpa/nori/blob/main/docs/blueprint-nori.md>

Os algoritmos avaliados — baseline, DQN e PPO — compartilham o mesmo modelo operacional, diferindo apenas na complexidade do processo de decisão. Enquanto a política baseline utiliza uma lógica simples, os algoritmos baseados em Aprendizado por Reforço Profundo realizam operações adicionais de inferência, o que se reflete em diferentes custos computacionais por ciclo de decisão. Para analisar a escalabilidade do processo de controle, foi introduzida uma carga computacional artificial no laço de decisão, controlada pelo parâmetro *work_units*. A variação desse parâmetro permite simular diferentes níveis de carga computacional e observar o impacto direto sobre o tempo de decisão de xApp.

Embora o modelo de decisão siga a estrutura típica de um agente de Aprendizado por Reforço, as ações geradas por meio de xApp não são aplicadas à dinâmica interna do simulador para alterar os KPIs da rede. Ou seja, o ambiente não atualiza seu estado em função das decisões tomadas. Desse modo, isola-se deliberadamente a análise do custo computacional do processo de decisão, sem avaliar os impactos das ações sobre os KPIs da rede. Isso possibilita analisar estritamente o overhead e a viabilidade temporal dos algoritmos no Near-RT RIC, mantendo fora do escopo deste trabalho a avaliação de desempenho da RAN em termos de QoS.

4.3. Configuração Experimental dos Algoritmos DRL

Os experimentos foram conduzidos comparando três abordagens de decisão executadas no xApp: uma política baseline, utilizada como referência, e dois algoritmos de Aprendizado por Reforço Profundo, DQN e PPO. Todas as abordagens foram avaliadas sob o mesmo modelo operacional, diferindo apenas no algoritmo utilizado para o processo de decisão.

O xApp foi configurado para operar com uma frequência fixa de execução do laço de decisão, mantida constante ao longo de todos os experimentos. Para avaliar o impacto da carga computacional no tempo de decisão, foi utilizado o parâmetro *work_units*, responsável por controlar a quantidade relativa de processamento executada a cada iteração. Foram considerados dois níveis de carga computacional, *work_units* igual a 200 e 800, representando cenários de baixa e alta carga computacional, respectivamente.

Os experimentos consideraram os perfis eMBB e URLLC como referência para os requisitos temporais das decisões de controle. Para cada combinação de algoritmo, perfil de serviço e carga computacional, o xApp foi executado por aproximadamente 60 segundos, de modo a coletar amostras suficientes para o cálculo das métricas estatísticas. Durante cada execução, foram registrados o algoritmo ativo, o valor de carga de trabalho (parâmetro *work_units*) e o tempo de decisão. Esses dados foram posteriormente armazenados em arquivos CSV e utilizados na análise comparativa do overhead computacional dos algoritmos avaliados.

4.4. Métricas de Desempenho

A métrica principal utilizada na avaliação experimental é o tempo de decisão de xApp, medido como o intervalo necessário para completar uma iteração do laço de decisão. Esse tempo inclui a execução do algoritmo de decisão configurado e representa o overhead computacional introduzido pelo processo de controle no Near-RT RIC. Essa métrica é particularmente relevante em ambientes Open-RAN, onde decisões precisam respeitar restrições temporais estritas, especialmente para serviços sensíveis à latência.

Para cada execução experimental, o tempo de decisão é registrado a cada iteração do laço e armazenado em arquivos de log no formato CSV. A partir desses registros, são extraídas estatísticas descritivas, incluindo média, mediana e percentis (p90, p95 e p99), com o objetivo de caracterizar não apenas o comportamento médio, mas também a variabilidade e os piores casos observados durante a execução.

Embora o ambiente de simulação considere perfis de serviço associados a eMBB e URLLC, os indicadores clássicos de desempenho da rede, como latência e vazão, permanecem constantes ao longo dos experimentos. Assim, esses indicadores não são utilizados como métricas comparativas, servindo apenas como referência para contextualizar os requisitos temporais associados a cada perfil de serviço. A análise concentra-se, portanto, na viabilidade temporal e na escalabilidade dos algoritmos de decisão quando executados como xApps no Near-RT RIC.

5. Resultados

Nesta seção, apresentamos os resultados obtidos a partir da execução de xApp sob diferentes configurações experimentais, considerando três algoritmos de decisão (baseline, DQN e PPO), dois níveis de carga computacional (200 e 800) e dois perfis de serviço (eMBB e URLLC). Cada execução teve duração aproximada de 60 segundos, durante os quais o tempo de decisão de xApp foi registrado a cada iteração do laço de controle.

A análise concentra-se no tempo de decisão de xApp como principal métrica de desempenho, permitindo avaliar o overhead computacional e a viabilidade temporal dos algoritmos quando executados no Near-RT RIC. As estatísticas apresentadas incluem medidas centrais e percentis elevados, de modo a capturar tanto o comportamento médio quanto os piores casos observados.

5.1. Comparação entre Algoritmos e Impacto da Carga Computacional

As Tabelas 1 e 2 apresentam a comparação quantitativa do tempo de decisão do xApp para os algoritmos baseline, DQN e PPO, considerando os níveis de carga de trabalho iguais a 200 e 800. Em ambos os perfis de tráfego, o baseline apresenta os menores valores médios e de 95º percentil (p95), refletindo sua baixa carga computacional.

Tabela 1. Estatísticas do tempo de decisão de xApp sob tráfego URLLC.

Algoritmo	Carga de trabalho	Média (ms)	Mediana (ms)	p95 (ms)	p99 (ms)
Baseline	200	0,316	0,282	0,589	0,905
Baseline	800	0,825	0,781	1,337	1,753
DQN	200	0,364	0,327	0,652	1,160
DQN	800	0,886	0,841	1,443	2,330
PPO	200	0,386	0,329	0,684	1,346
PPO	800	0,927	0,902	1,437	1,709

Tabela 2. Estatísticas do tempo de decisão de xApp sob tráfego eMBB.

Algoritmo	Carga de trabalho	Média (ms)	Mediana (ms)	p95 (ms)	p99 (ms)
Baseline	200	0,312	0,275	0,589	0,982
Baseline	800	0,847	0,820	1,366	1,656
DQN	200	0,355	0,322	0,616	0,973
DQN	800	0,872	0,837	1,490	1,974
PPO	200	0,386	0,325	0,723	1,927
PPO	800	0,890	0,875	1,398	1,998

O aumento da carga computacional de 200 para 800 resulta em crescimento consistente do overhead para todos os algoritmos, sendo esse efeito mais pronunciado para DQN e PPO. Entre os métodos baseados em aprendizado por reforço profundo, o PPO apresenta os maiores valores nos percentis superiores, indicando maior sensibilidade ao aumento da carga. Apesar dessas diferenças, os tempos observados permaneceram compatíveis com os requisitos temporais do Near-RT RIC, permanecendo abaixo do limite superior de 1 segundo normalmente associado ao domínio Near-RT do Open-RAN. Esses resultados indicam compatibilidade temporal dos algoritmos avaliados com a execução como xApps no Near-RT RIC.

As Figuras 1 e 2 apresentam o tempo de decisão de xApp em função da carga computacional artificial para os algoritmos baseline, DQN e PPO. Em ambos os perfis de tráfego, constata-se aumento consistente do tempo de decisão em função da elevação da carga computacional avaliada (de 200 para 800). Esse comportamento indica influência da complexidade do algoritmo sobre o custo computacional do processo de decisão.

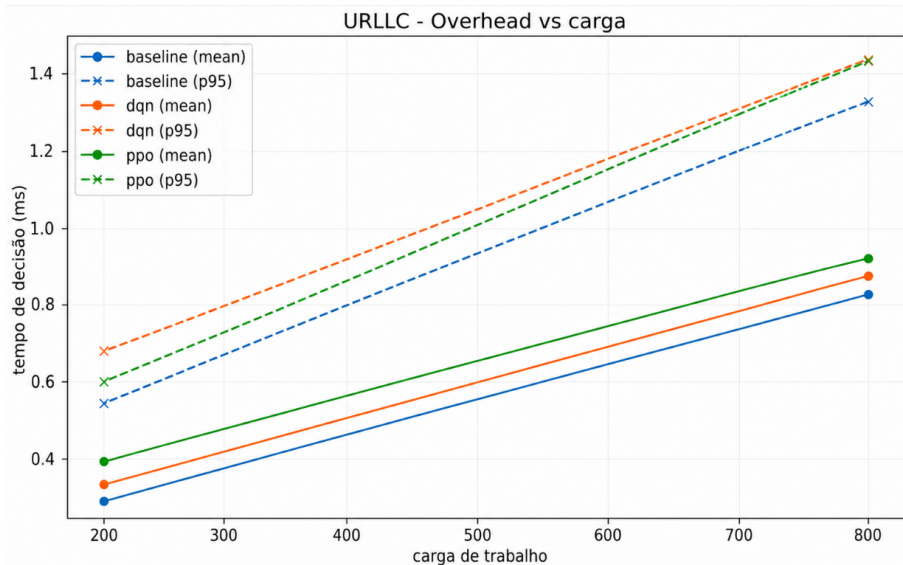


Figura 1. Overhead sob tráfego URLLC.

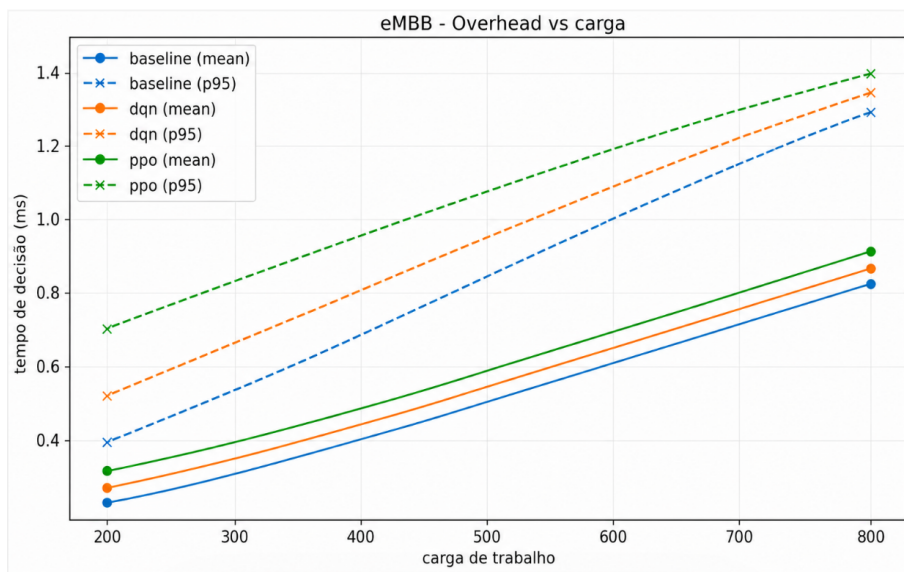


Figura 2. Overhead sob tráfego eMBB.

5.2. Análise do Overhead sob Tráfego URLLC

No cenário URLLC, o comportamento observado nos gráficos de linha é complementado pela análise da distribuição do tempo de decisão apresentada nos diagramas de violino da Figura 3. O algoritmo baseline apresenta não apenas o menor tempo médio de decisão, mas também a menor dispersão dos valores, indicando maior previsibilidade temporal.

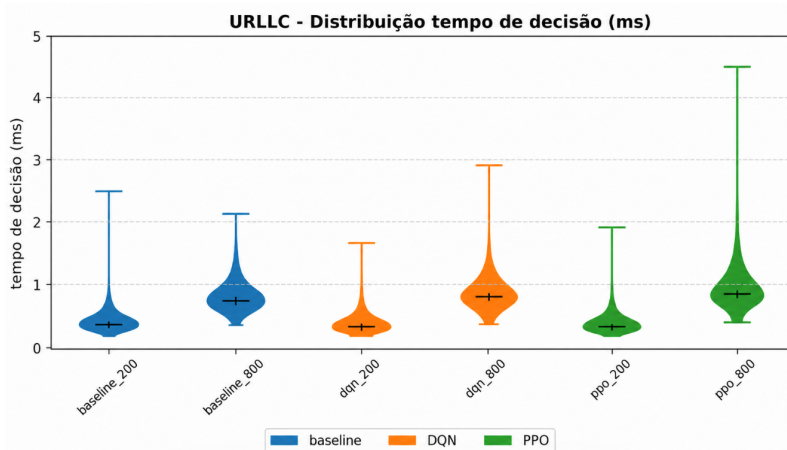


Figura 3. Distribuição do tempo de decisão sob tráfego URLLC.

Os algoritmos DQN e PPO apresentam maior variabilidade, especialmente no cenário de maior carga computacional ($work_units = 800$), onde observa-se o aumento do intervalo interquartil e a presença de valores extremos. Esse comportamento é confirmado pelos valores estatísticos apresentados na Tabela 1, em que os percentis elevados (p95 e p99) crescem de forma mais acentuada para os métodos baseados em aprendizado por reforço. Apesar desse aumento de variabilidade, os tempos médios de decisão permanecem abaixo de 1 segundo em todos os cenários avaliados. Esse resultado indica que,

mesmo sob maior carga computacional, os algoritmos DQN e PPO mantiveram compatibilidade temporal com a execução como xApps no Near-RT RIC.

5.3. Análise do Overhead sob Tráfego eMBB

No cenário eMBB, os resultados seguem tendência semelhante à observada para URLLC. Os gráficos de linha indicam crescimento consistente do tempo de decisão de xApp à medida que a carga computacional aumenta, mantendo o baseline como o algoritmo de menor overhead, seguido pelo DQN e pelo PPO.

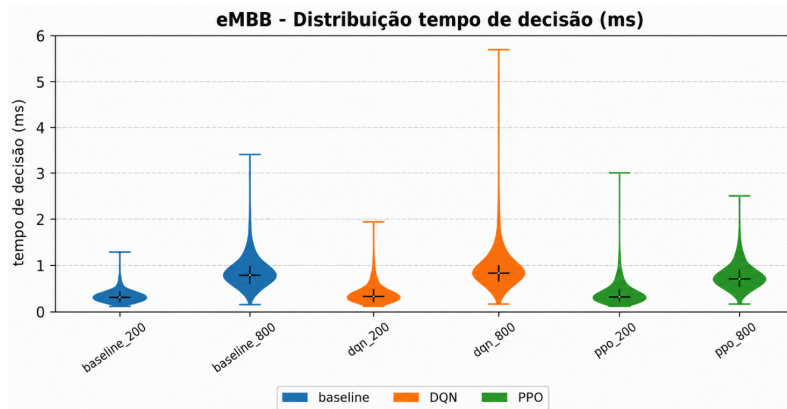


Figura 4. Distribuição do Tempo de Decisão sob tráfego eMBB.

A análise dos diagramas de violino da Figura 4 mostra que, embora as médias sejam semelhantes às do cenário URLLC, os algoritmos DQN e PPO apresentam maior dispersão e mais valores extremos sob cargas computacionais elevadas. Esses efeitos, observados nos percentis superiores da Tabela 2, indicam maior sensibilidade a flutuações ocasionais no tempo de decisão. Ainda assim, os tempos permanecem compatíveis com laços de controle do Near-RT RIC, reforçando a viabilidade do uso de aprendizado por reforço em ambientes Open-RAN.

5.4. Discussão dos Resultados

Os resultados mostram que o overhead computacional de xApp aumenta de forma consistente com a elevação da carga computacional artificial (work units), tanto para URLLC quanto para eMBB. Esse comportamento confirma que a complexidade do processo de decisão influencia diretamente o tempo de execução da xApp, independentemente do perfil de tráfego. Em todos os cenários avaliados, o algoritmo baseline apresentou o menor overhead, seguido pelo DQN e, por fim, pelo PPO, refletindo o maior custo computacional observado para o PPO no cenário experimental avaliado. Ainda assim, mesmo nos cenários mais exigentes (800 work units), os tempos médios e os percentis elevados permaneceram abaixo de 1 segundo, indicando que tanto DQN quanto PPO são viáveis para execução no Near-RT RIC sob o ponto de vista temporal.

Estes achados corroboram a literatura recente focada na avaliação de algoritmos DRL em arquiteturas Open-RAN, a exemplo de [Bordin et al. 2025], os quais também reportam maior overhead em PPO quando comparado a métodos off-policy. Diferentemente de trabalhos que avaliam ganhos diretos de QoS, este estudo isola o custo computacional

do processo de decisão, evidenciando que os resultados indicam compatibilidade temporal do uso de DRL em xApps mesmo sob cargas elevadas. As evidências sugerem que o principal trade-off associado ao uso de DRL no Near-RT RIC está relacionado ao aumento do custo computacional em troca de maior flexibilidade do processo decisório.

6. Conclusão

Este trabalho analisou o overhead computacional de algoritmos de Aprendizado por Reforço executados como xApps no Near-RT RIC em um ambiente Open-RAN baseado no Blueprint Nori v1. Os resultados mostraram aumento consistente do tempo de decisão com a elevação da carga computacional, sendo o baseline o algoritmo de menor overhead, seguido por DQN e PPO. Apesar do maior custo computacional dos métodos baseados em DRL, os tempos observados permaneceram compatíveis com os requisitos temporais do Near-RT RIC, indicando compatibilidade temporal desses algoritmos com a execução como xApps no Near-RT RIC.

Como trabalhos futuros, pretende-se estender a análise para ambientes com controle em ciclo fechado, nos quais as ações de xApp impactem diretamente os KPIs da rede, permitindo avaliar conjuntamente overhead computacional e desempenho de QoS.

Agradecimentos

Os autores agradecem à agência de fomento FUNCAP pelo suporte financeiro, bem como aos revisores anônimos pelas valiosas contribuições para o aprimoramento deste trabalho.

Referências

- Alam, K., Habibi, M. A., Tammen, M., Krummacker, D., Saad, W., Renzo, M. D., Melodia, T., Costa-Pérez, X., Debbah, M., Dutta, A., and Schotten, H. D. (2026). A comprehensive tutorial and survey of o-ran: Exploring slicing-aware architecture, deployment options, use cases, and challenges. *IEEE Communications Surveys Tutorials*, 28:1637–1678.
- Bansbach, E.-M., Eliachevitch, V., and Schmalen, L. (2021). Deep reinforcement learning for wireless resource allocation using buffer state information. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE Press.
- Bordin, M., Lacava, A., Polese, M., Satish, S., Nittoor, M. A., Sivaraj, R., Cuomo, F., and Melodia, T. (2025). Design and evaluation of deep reinforcement learning for energy saving in open ran. In *2025 IEEE 22nd Consumer Communications Networking Conference (CCNC)*, pages 1–6.
- Chen, G., Shao, R., Shen, F., and Zeng, Q. (2023). Slicing resource allocation based on dueling dqn for embb and urllc hybrid services in heterogeneous integrated networks. *Sensors*, 23(5):2518.
- Hussien, O. and Jahankhani, H. (2025). Optimizing 5g resource allocation in pso with machine learning approach to open ran architectures. *Journal of Next-Generation Research* 5.0.
- Menezes, D., Medeiros, D., Couto, R., Caminha, P., Souza, L., Táparo, F., Thomaz, G., Valle, J., Oliveira, F., Campista, M., Costa, L., and Moraes, I. (2023). *Ameaças e Vulnerabilidades em Open RAN: Desafios e Soluções*, pages 150–201.

- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.
- O-RAN Software Community (O-RAN SC) (2025). O-ran sc non-rt ric project – nonrttric master documentation. Available at: <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrttric/en/latest/index.html>.
- Park, H., Seol, K., Song, S., Lee, Y., and Park, L. (2024). Recent research on reinforcement learning for open ran. In *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1744–1745.
- Polese, M., Bonati, L., D’Oro, S., Basagni, S., and Melodia, T. (2023). Colo-ran: Developing machine learning-based xapps for open ran closed-loop control on programmable experimental platforms. *IEEE Transactions on Mobile Computing*, 22(10):5787–5800.
- Singh, S. K., Singh, R., and Kumbhani, B. (2020). The evolution of radio access network towards open-ran: Challenges and opportunities. In *2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6.
- Sohaib, R. M., Shah, S. T., Jamshed, M. A., Onireti, O., and Yadav, P. (2025). Optimizing urllc in open ran: A deep reinforcement learning-based trade-off analysis. *IEEE Communications Standards Magazine*, 9(3):33–39.
- Soltani, S., Amanloo, A., Shojafar, M., and Tafazolli, R. (2025). Intelligent control in 6g open ran: Security risk or opportunity? *IEEE Open Journal of the Communications Society*, 6:840–880.
- Thieu, H.-T., Pham, V.-Q., Kak, A., and Choi, N. (2023). Demystifying the near-real-time ric: Architecture, operations, and benchmarking insights. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–8.
- Tsampazi, M., D’Oro, S., Polese, M., Bonati, L., Poitau, G., Healy, M., and Melodia, T. (2023). A comparative analysis of deep reinforcement learning-based xapps in o-ran. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pages 1638–1643.
- Zheng, J. (2025). Research on deep reinforcement learning-based resource allocation strategies for wireless networks. In *Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence, DEAI ’25*, pages 170–176, New York, NY, USA. Association for Computing Machinery.