

Handover em redes aéreas 5G: Uma solução baseada em aprendizado por reforço

Maria Barbosa¹, Marcelo Batista¹, Anderson Queiroz¹, David Cavalcanti¹, Kelvin Dias¹

¹Centro de Informática – Universidade Federal de Pernambuco (CIn - UFPE)
Recife – PE – Brazil

{mksb, mvbs3, aalq, djmc, kld}@cin.ufpe.br

Resumo. *Os Veículos Aéreos Não Tripulados (UAV) como estações rádio-base, servindo a usuários terrestres, vem ganhando tração com as novas releases 5G e futuros sistemas 6G. Fornecendo conectividade ubíqua em áreas remotas, desassistidas pelas operadoras ou zonas rurais. Contudo, manter a continuidade do serviço em redes UAV, especialmente durante o handover, é mais desafiador que nas redes terrestres. Isso ocorre devido à menor cobertura da rede, aumentando o risco de efeito ping-pong. Portanto, este artigo propõe uma solução baseada em aprendizado por reforço, que utiliza contexto de mobilidade do usuário e da rede. Os resultados indicam a eficácia da proposta, com uma redução de 74% nas falhas de handover em comparação com soluções da literatura.*

Abstract. *Unmanned Aerial Vehicles (UAV) as Base Stations, serving ground users, has been gaining momentum with the new 5G releases and in future 6G systems. Providing ubiquitous connectivity in remote, underserved, or rural areas. However, ensuring service continuity in UAV networks, especially during handover, is more challenging than in ground cellular networks. This is due to the smaller network coverage, increasing the risk of the ping-pong effect. Therefore, this article proposes a solution based on reinforcement learning, which utilizes user mobility and network contexts. The results indicate the effectiveness of the proposal, with a 74% reduction in handover failures compared to state-of-the-art solutions.*

1. Introdução

Os Drones ou Veículos Aéreos Não Tripulados, do inglês *Unmanned Aerial Vehicles* (UAVs), são produtos comerciais prontos para uso (*Commercial off-the-Shelf* - COTS) que incorporam e integram tecnologias de rádio e recursos computacionais. Esses UAVs oferecem suporte para superar os desafios das comunicações móveis modernas [Alsoliman et al. 2021]. Tais características tornam os UAVs atrativos para uma ampla variedade de aplicações práticas, incluindo sistemas de entregas de encomendas, transmissão em tempo real para entretenimento, cobertura de notícias, monitoramento agrícola, segurança passiva e ativa, busca e salvamento [Mishra and Natalizio 2020].

À medida que as redes móveis celulares evoluem com os sucessivos lançamentos de padrões do 3rd *Generation Partnership Project* (3GPP), também avançam o suporte dessas redes quanto à conectividade e comando/controle de UAVs. Esse suporte aos UAVs começa a ser mais evidenciado com a última *release* do 4G/*Long Term Evolution* e com a *release* 15 do *New Radio/5G*, a primeira da quinta geração das redes móveis celulares

[Muruganathan et al. 2021]. O 3GPP, com o 5G-Advanced a partir da *release* 18, apresenta melhorias na eficiência de transmissão, identificação e sinalização para UAVs, oferecendo maior suporte para diferentes arquiteturas e modos de comunicação [Lin 2022].

Recentemente, os UAVs também estão assumindo a função de estações rádio-base aéreas, denominadas UAV-BS, viabilizando conectividade a usuários terrestres independentemente da infraestrutura celular tradicional [Derhab et al. 2023]. Espera-se que nas redes 5G atuais e nos futuros sistemas 6G, os UAV-BSs formem redes aéreas para fornecer acesso sem fio ubíquo em áreas remotas, não atendidas pelas operadoras ou zonas rurais. Os UAV-BSs também devem proporcionar conectividade sem fio confiável para equipes de resgate em cenários de desastre, para carros conectados/autônomos e para os Sistemas de Transporte Inteligente [Tafintsev et al. 2023]. Além disso, UAV-BSs estendem e melhoram o sinal na borda da cobertura celular terrestres, por reduzirem a distância entre usuários e pontos de acesso, criando enlace com linha de visada (*Line-of-Sight* - LoS) que melhoram significativamente a qualidade do sinal [Gangula et al. 2018].

Assim como toda tecnologia emergente, as redes baseadas em UAV ainda enfrentam desafios significativos como a área de cobertura reduzida, posicionamento das estações-rádio base e operação por tempo limitado devido à autonomia das baterias [Aydin et al. 2021]. Diante dessas limitações e da natureza dinâmica dos dispositivos conectados, surgem problemas adicionais, como prover *handover* transparente para os equipamentos do usuário (*User Equipment* - UEs) terrestres servidos por UAV-BSs. Assim, garantir os requisitos de qualidade de serviço (QoS), especialmente durante o *handover*, é desafiador. As soluções atuais priorizam a trajetória e predição de sinal, negligenciando as condições da rede e do usuário [Yang et al. 2017][Hu et al. 2019][Queiroz et al. 2023]. Este trabalho é motivado pela necessidade de compreender as peculiaridades das redes UAVs e desenvolver uma estratégia eficaz para melhorar o *handover* em tais ambientes.

Neste contexto, este artigo propõe um mecanismo baseado em contexto utilizando aprendizado por reforço profundo, o *Double Deep Q-Network* (DDQN), para aprimorar o procedimento de *handover*. A solução proposta utiliza uma combinação de parâmetros de mobilidade e de redes para determinar a necessidade de realização de *handover*. A proposta é comparada com diferentes mecanismos baseados no modelo *default*, definidos pelo 3GPP, e mecanismos baseados em aprendizado supervisionado. A proposta e os trabalhos relacionados são avaliados em cenários distintos de mobilidade.

As próximas seções foram estruturadas da seguinte forma: a Seção 2 traz os estudos relacionados às soluções inteligentes de *handover* em redes aéreas. A Seção 3 apresenta a proposta com seus componentes arquiteturais e detalha o mecanismo de *handover* baseado em DDQN. A Seção 4 apresenta a caracterização da etapa de treinamento e elabora os cenários de mobilidade. A Seção 5 descreve o ambiente de simulação. A Seção 6 aborda os resultados dos mecanismos de *handover* e, por fim, as conclusões e trabalhos futuros são apresentados na Seção 7.

2. Trabalhos Relacionados

Esta seção apresenta o estado atual das soluções de *handover* baseadas em aprendizado de máquina, expondo diferentes propostas que buscam otimizar o processo de tomada de decisão deste importante mecanismo de mobilidade. Em cenários envolvendo UAV-BS's, os possíveis erros de execução de *handover* (efeito ping-pong ou não realização) tendem a

umentar, uma vez que a área de cobertura de uma rede UAV é menor quando comparada a de uma rede móvel convencional. O incremento do nível de interferência entre as células vizinhas, exigem do mecanismo mais eficiência e precisão sobre qual momento tal ação deve ser executada.

A proposta de [Yang et al. 2017] utiliza Redes Neurais Recorrentes (RNN) no intuito de prever trajetórias com base em dados obtidos da geolocalização, como latitude e longitude. Essas previsões são usadas para determinar a intensidade do sinal e a necessidade da realização de *handover*. O modelo é treinado e validado com o banco de dados de trajetórias GPS *GeoLife*. No entanto, o RNN enfrenta o problema de dissipação/explosão do gradiente durante o treinamento, dificultando a atualização dos pesos das camadas mais antigas e afetando a aprendizagem de dependências de longo prazo.

Em [Hu et al. 2019] também é apresentado um modelo de previsão de trajetória do UE, contudo, os autores utilizam *Gated Recurrent Unit* (GRU) e *social pooling*. Esse método considera os movimentos dos vizinhos em áreas lotadas ao prever o deslocamento do UE, incorporando o *social pooling* para melhorar a precisão do mecanismo. No entanto, sendo as previsões realizadas pela UAV-BS e isto, pode levar a um consumo de energia significativo, reduzindo o tempo de operação do UAV.

Em [Queiroz et al. 2023], é realizada uma análise comparativa de modelos de aprendizado supervisionado, incluindo RNN, GRU e *Long Short Term Memory* (LSTM), para prever a potência do sinal de rádio das UEs em solo e auxiliar em decisões de *handover*. O estudo conclui que o modelo mais adequado para cenários de BS em UAV é o GRU para prever a potência do sinal. No entanto, o trabalho carece de diferentes cenários de mobilidade para os usuários, que são considerados sempre em um único sentido de deslocamento, e não utiliza a combinação de parâmetros de localização e de rede para formular a ação do mecanismo.

Em [Jang et al. 2022], é proposto um modelo baseado em Deep Q-Network (DQN) junto com um algoritmo de Otimização de Política Proximal para reduzir o número de handovers desnecessários, desconsiderando flutuações de RSSI, e otimizar políticas de atribuição de recompensas para diminuir o consumo de energia. No entanto, os handovers são realizados utilizando os UAVs como usuários finais e não como estações base de comunicação.

A partir do levantamento do estado da arte apresentado, identificou a inexistência de artigos na atual literatura que levam em consideração o contexto do usuário terrestre numa rede UAV juntamente com técnicas otimizadas de aprendizado por reforço profundo para a determinação da realização de *handover*, que é o foco principal deste artigo. A presente proposta captura o contexto do usuário, em diferentes cenários, tanto do ponto de vista de localização e mobilidade quanto do ponto de vista da rede.

3. Proposta

Esta seção apresenta os componentes arquiteturais e o mecanismo de *handover* em redes aéreas montadas em UAV proposto neste artigo. O mecanismo visa otimizar o *handover*, utilizando aprendizado por reforço profundo e incorporando o contexto do usuário, tais como informações de mobilidade/localização e parâmetros de redes. A arquitetura apresentada na Figura 1 é composta por dois blocos principais: (a) A aplicação de contexto

localizada no dispositivo móvel do usuário, (b) O mecanismo de contexto dividido em duas partes, uma localizada na borda da rede e outra localizada no núcleo da rede.

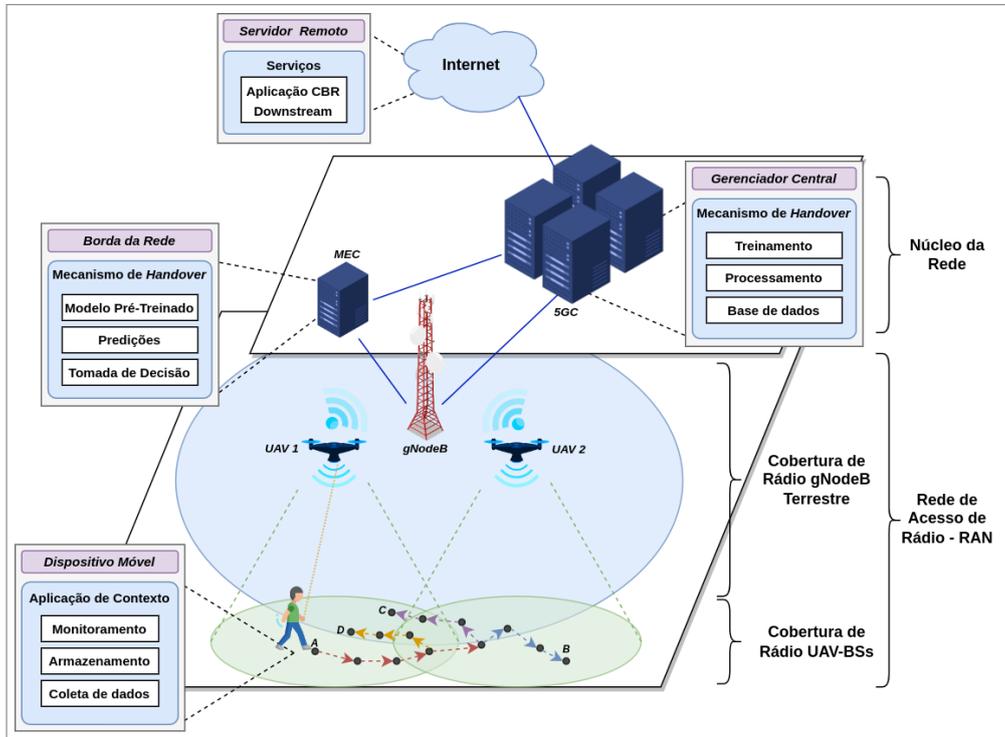


Figura 1. Visão Geral da Arquitetura Proposta.

O dispositivo móvel do usuário possui uma aplicação que captura os dados contextuais necessários para a proposta. Essas informações são enviadas para o gerenciador central, sendo armazenadas numa base de dados, e posteriormente, utilizadas para treinar o modelo de *Machine Learning* (ML). Vale ressaltar que toda a etapa de treinamento é realizado no gerenciador central, e uma vez que o modelo é treinado, é enviado para a borda da rede. Enquanto isso, a parte do mecanismo localizada na borda da rede é responsável por armazenar o modelo pré-treinado e, ao receber o estado do usuário, realizar previsões e determinar se a ação a ser realizada é a execução do *handover*. Por fim, o fornecedor de serviços possui uma aplicação de Taxa de Bits Constante (*Constant Bitrate - CBR*), que permite ao usuário realizar *downstream* a uma taxa de transferência de 5,5 Mbps, correspondendo à transmissão de um vídeo com resolução HD 1080p. Essa aplicação foi utilizada para gerar tráfego na rede, permitindo a captura das métricas de QoS.

3.1. Mecanismo baseado em Aprendizado por Reforço Profundo

O mecanismo proposto reúne informações de mobilidade, como localização (Loc_{ue}) e velocidade (Spd_{ue}), juntamente com métricas de rede, incluindo taxa de acertos na entrega de pacotes (*hit rate*), Indicador de Qualidade do Canal (CQI) e Indicador de Força do Sinal Recebido (RSSI) da UAV-BS à qual o usuário está conectado e das vizinhas. Esses dados são utilizados para determinar o contexto do usuário, aprimorando assim a precisão no processo de tomada de decisão da realização de *handover*. Dessa forma, visando empregar técnicas que assegurem uma otimização na utilização dessas informações, propõe-se aplicar algoritmos de aprendizado por reforço profundo. Esse tipo de aprendizagem recebem o estado (contexto) e, a partir desse agente, toma uma ação e é recompensado.

O Algoritmo 1 apresenta o funcionamento do mecanismo de *handover*. A aplicação de contexto, localizada no dispositivo móvel, obtém o estado do usuário a cada 200 milissegundos, encaminhando-o para a borda da rede. Após receber o estado (S_t), o modelo pré-treinado é carregado para verificar a necessidade de execução do *handover*. Após a realização da predição, o modelo envia uma ação (A_t), realizar ou não o *handover* para o usuário. Se a A_t prevista for executar o *handover*, o usuário inicia o processo e, após a conclusão, transmite a recompensa de volta para o agente. Posteriormente, a aplicação de contexto reagenda a captura de estado, e aguarda a próxima A_t .

Algorithm 1: Mecanismo de *handover* sensível ao contexto.

Input: $Modelo, Loc_{ue}, Spd_{ue}, HitRate, CQI, RSSI_n$
Function GerenciadorHandover:
 $S_t = [Loc_{ue}, Spd_{ue}, HitRate, CQI, RSSI_{att}, RSSI_{tgt}];$
 $A_t = Modelo.predict(S_t);$
if $ExisteCandidato(A_t)$ **then**
 $FazerHandover(UAVBS_{Alvo-id});$
 $Agendar(200ms, Recompensa());$
 $Agendar(200ms, GerenciadorHandover());$

3.2. Modelo DDQN

O modelo utilizado nesta proposta foi o *Double Deep Q-Network* (DDQN) [Hasselt et al. 2016], um algoritmo que se baseia na abordagem tradicional *Deep Q-Network* (DQN). Contudo, ele supera uma das limitações do DQN original, que é a superestimação dos valores de ação. Ao introduzir uma rede de destino e uma rede online separada, o algoritmo DDQN fornece estimativas de valor mais estáveis e precisas. A rede de destino, que atua como o crítico, é uma cópia separada da rede online que é periodicamente atualizada com os pesos da rede online, sendo usada para estimar os valores Q para o próximo estado no processo de treinamento.

Os hiperparâmetros utilizados para modelar o DDQN estão apresentados na Tabela 1. Esses valores foram determinados por meio da técnica de otimização denominada *Grid Search*, sendo feita uma validação cruzada para determinar a configuração que proporcionaria o melhor desempenho.

Tabela 1. Hiperparâmetros selecionados para a elaboração do DDQN.

<i>Input Layer</i>	<i>Hidden Layer</i>	<i>Output Layer</i>	<i>Optimizer</i>	<i>Activation Function</i>
Dense - (256, 7)	Dense - 256	Dense - 2	Adam	Relu

A formulação da recompensa pelo usuário abrange três métricas principais: $\Delta RSSI$, CQI e HitRate. O $\Delta RSSI$ representa a diferença na intensidade do sinal entre a célula atual do usuário e as células vizinhas, enquanto o CQI indica a qualidade do canal. Essas características orientam a identificação de momentos ideais para *handover*, especialmente quando os valores $\Delta RSSI$ e CQI diminuem. Além disso, *handovers* frequentes são indesejáveis devido ao potencial de perda de pacotes e para lidar com essa situação, a métrica Hitrate é utilizada. Quanto mais o agente realiza *handovers*, menor

torna-se a taxa de acerto, refletindo um aumento na perda de pacotes durante o processo e destacando a necessidade de uma abordagem equilibrada.

A Tabela 2 exibe as equações que regem recompensas positivas e negativas para cada métrica mencionada anteriormente. Recompensas positivas são concedidas ao agente por escolhas de ações corretas, e as equações são ajustadas para manter as recompensas em torno de 100. Em contraste, para recompensas negativas, as equações foram elaboradas para resultar em penalidades mais severas. Quando o agente toma uma decisão incorreta, a penalização é mais acentuada, enfatizando a gravidade de ações incorretas.

Tabela 2. Equações de penalização positiva e negativa utilizada no Agente DDQN.

Métrica	Recompensa Positiva	Recompensa Negativa
Δ RSSI	$100 \cdot \tanh(\Delta \cdot 0.05)$	$-16\Delta^2 - 32\Delta$
HitRate	$100 \cdot \tanh(26.4 \cdot \text{hitRate} - 2637)$	$-0.15 \cdot (\text{hitRate}^2) - 3.2 \cdot \text{hitRate}$
CQI	$100 \cdot \tanh(13.2 \cdot \text{cqi} - 195)$	$2030 \cdot \text{cqi} - 30080$

4. Treinamento do Modelo DDQN

A etapa de treinamento realizada pelo Gerenciador Central utilizou um ambiente de simulação. Visando ter um melhor resultado no treinamento, foram realizadas 180 simulações com duração de 100 segundos e os cenários de mobilidade descritos a seguir.

4.1. Cenários de Mobilidade

Visando explorar os aspectos críticos e desafiadores para a decisão de handover, como a mobilidade que pode propiciar o efeito ping-pong, são considerados três cenários distintos, cada um retratando diferentes movimentos de usuários dentro dos limites da célula.

1. **Cenário 1:** O usuário se desloca em direção à próxima célula, representada pela linha tracejada azul na Figura 1, até chegar ao ponto **B**. Neste cenário, o dispositivo entra e permanece na célula vizinha, resultando em um único *handover*.
2. **Cenário 2:** O usuário realiza uma entrada na célula adjacente e retorna para a célula inicial. A trajetória dada pela linha roxa na Figura 1, dirigindo-se até o ponto **C**. Neste cenário, o *handover* é executado duas vezes, uma quando o usuário entra na cobertura da célula vizinha, outro quando retorna à célula de origem.
3. **Cenário 3:** Deslocamento até a fronteira da célula atual, seguido de um retorno, em vez de atravessar para a célula vizinha. A Figura 1 apresenta o usuário movendo-se até o ponto **D**, linha laranja. Neste cenário não existe *handover*.

4.2. Geração do Conjunto de Dados

A partir dos cenários propostos, deu-se início a concepção de movimentos aleatórios para o usuário. Inicialmente, o SUMO¹ foi considerado para a gerar os deslocamentos, contudo, observou-se que não haveria garantias de caracterização dos cenários propostos, tornado-se necessária a criação de uma nova metodologia para a geração de dados aleatórios. Sendo assim, foram definidos os número de coordenadas necessárias para o treinamento, os valores mínimos que representam a localização inicial do usuário e o valor máximo que representa a localização limite do usuário. Em seguida, as coordenadas x

¹<https://eclipse.dev/sumo/>

e y são inicializadas com os valores mínimos e, posteriormente, ocorre uma iteração pelo número especificado de coordenadas.

Para o cenário 1, o usuário possui um movimento contínuo em direção à célula adjacente, sendo assim, o percurso não possui limite de deslocamento, portanto, os valores da coordenada x são incrementados aleatoriamente em cada iteração até todas as coordenadas serem definidas. Por outro lado, os cenários 2 e 3 possuem limites de deslocamento, sendo o do cenário 2 o maior, uma vez que o usuário entra na célula vizinha. Em ambos os casos, a condição inicial é semelhante à do cenário 1, contudo, se x ultrapassar a fronteira ou a metade das coordenadas for atingida, é iniciado o ajuste em x para a direção oposta, simulando um retorno em direção à célula inicial. A velocidade do usuário é uma função das coordenadas x e y , estando na faixa de 0,5 m/s até 3 m/s. Esses valores correspondem a UEs em uso por pedestres e em veículos com baixa velocidade (por exemplo, passageiros em veículos em regiões congestionadas).

4.3. Resultados do Treinamento

Os resultados do treinamento são apresentados na Figura 2. Nota-se nas simulações iniciais a fase exploratória, onde diferentes estratégias são testadas buscando identificar aquelas que resultam em recompensas mais favoráveis, resultando em cerca de 275 *handovers* por simulação. À medida que o agente interage repetidamente com o ambiente, espera-se que seu desempenho evolua, adaptando-se para otimizar a execução do *handover* com base nas informações acumuladas durante o processo de aprendizagem. Sendo assim, nota-se que por volta da quinquagésima simulação, o agente começa a ajustar seu comportamento conforme as recompensas e penalidades associadas às suas ações, passando a realizar *handover* apenas quando necessário.

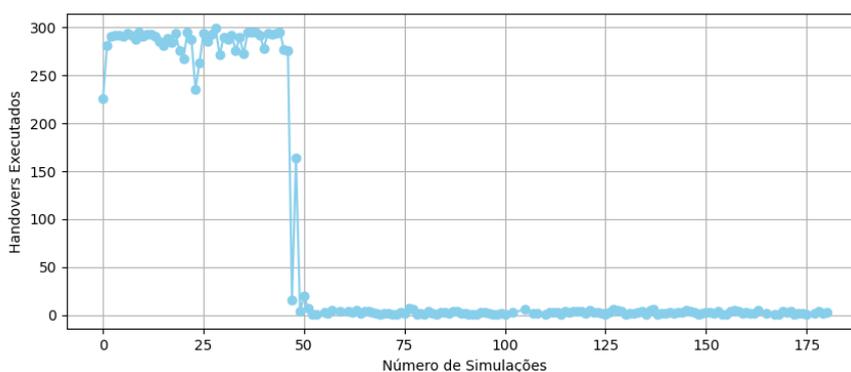


Figura 2. Número de *handover* executando ao longo do processo de treinamento.

5. Simulando Redes Aéreas

A implementação da rede aérea composta por UAV-BSs, rede terrestre, do mecanismo proposto e das soluções do estado da arte no estudo comparativo, foi baseada em ferramentas de simulação de eventos discretos. Essas ferramentas incluem o OMNeT++² e o Simu5G³. Para desenvolver adequadamente a simulação para esta proposta, essas

²<https://omnetpp.org/>

³<http://simu5g.org/>

ferramentas e bibliotecas foram integradas para representar melhor o objeto desta pesquisa, destacando assim resultados obtidos por modelos de simulação representativos de ambientes reais. No entanto, o Simu5G não inclui o 5G aéreo, então foi necessário implementar as características da rede UAV. Dessa forma, a gNodeB existente no Simu5G foi adaptada com funcionalidades ligadas ao UAV como a adição do eixo Z, bem como a implementação de um modelo de canal (*air-ground*) 5G [Queiroz et al. 2023]. A Tabela 3 apresenta os valores dos parâmetros adotados na simulação.

Tabela 3. Parâmetros utilizados na simulação.

Frequência da Portadora (f_c)	3,5 GHz	Número de UEs	1
UAV-BS's txpower	28dBm	Altura do UAV	50 metros
Número de UAVs	2	Raio da Célula	150 metros

6. Avaliação da Proposta

Esta seção apresentará os resultados obtidos pelo mecanismo de *handover*. Os mecanismos utilizados para a comparação com a proposta incluem: o modelo 3GPP de *handover* baseado em histerese, nomeado de *Default*; e um mecanismo baseado em predição de sinal utilizando o GRU [Queiroz et al. 2023]. Para realizar essa comparação, serão avaliados o quão próximas às soluções estão do momento ótimo para a realização e as taxas de erros de execução (efeito ping-pong), além dos impactos nas métricas de QoS, como atraso médio e perda de pacotes durante *handover*.

Para a realização das simulações de validação, foram gerados dados de testes, conforme apresentado na Seção 4.2, e importados no OMNeT++. Em todos os cenários, as configurações de rede 5G SA utilizadas segundo a Seção 5. Para alcançar um intervalo de confiança de 95%, foram realizadas cento e cinquenta simulações para cada modelo. O tráfego gerado pelo Servidor Remoto destinados as UEs terrestres, cujos parâmetros são indicados na Seção 3, possibilitou a obtenção das métricas de QoS.

A Figura 3 apresenta os resultados para a execução de *handover* para os três cenários avaliados. Esse valor corresponde ao tempo de simulação no qual essa ação é realizada, o disparo antecipado ou retardado pode causar o efeito ping-pong, conforme evidenciado pela Figura 4. As linhas tracejadas representam o intervalo ideal no qual a UE teria para realizar o *handover*. Esses valores foram obtidos pela análise do comportamento do RSSI do dispositivo móvel em relação aos UAV-BSs, no momento em que a potência da célula vizinha ultrapassou a da célula servidora. A linha verde corresponde ao valor ótimo para a realização do *handover*, enquanto a linha vermelha indica o momento máximo ideal.

Conforme apresentado na Figura 3, para o Cenário 1 - Célula 3, o GRU obteve uma redução de 6,2% em relação ao valor ótimo, enquanto o *Default* teve um aumento de 7,8% sobre o tempo máximo. Em contrapartida, o DDQN apresentou um valor médio dentro do intervalo ótimo, com um erro de cerca de 4% maior que o valor ótimo, indicado pela linha verde.

Para o Cenário 2, a Figura 3 apresenta os resultados divididos em duas partes: o primeiro *handover* é realizado para a célula 3, tendo uma margem de segurança reduzida,

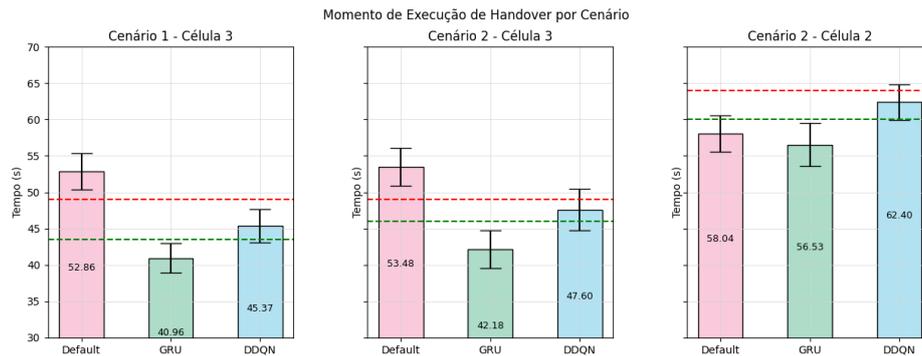


Figura 3. Momento da simulação quando o *handover* foi disparado.

devido ao retorno do usuário para a célula 2. Na primeira parte, percebe-se um comportamento similar ao cenário 1, pois o *Default* ultrapassa o valor máximo em cerca de 9,20% e o GRU exibe uma redução de cerca de 8,3% menor que o valor ótimo. Na análise do retorno para a célula 2, o modelo *GRU* mostrou a maior distância em relação ao momento ideal, em cerca de 5,8% menor. Para o modelo *Default* nota-se uma redução de cerca de 3,2% em relação ao valor ideal, além disso, observa-se a proximidade entre os valores médios do primeiro e do segundo *handover*, podendo causar o efeito *ping-pong*. O DDQN, por sua vez, permanece dentro da faixa de segurança para ambos os casos, com uma pequena parte do intervalo de confiança fora dos limites ideais. Sendo assim, fica evidenciado que o *Default* tende a retardar atuação, devido à demora para satisfazer a histerese, enquanto o GRU tende a antecipar a decisão devido a uma limitação do modelo por apenas avaliar a previsão de sinal e não compreender o contexto do usuário. No cenário 3 não há *handovers* e por isso não houve um gráfico para este cenário.

Para ter-se uma visão completa da eficácia do procedimento de *handover* não basta apenas analisar os tempos de atuação, mas também o número execuções. Posto isso, a Figura 4 mostra o número de *handovers* em cada cenário. No cenário 1 espera-se que cada mecanismo realize apenas um *handover*. A figura evidencia a aproximação do DDQN da quantidade esperada de *handovers*, apresentando apenas uma execução em excesso. Enquanto, o GRU resultou em 3 execuções em excesso e 6 falhas, ou seja, não havendo a realização do *handover* quando deveria ter acontecido. O modelo *Default* obteve o pior resultado, além de executar o primeiro *handover* 150 vezes, realizou 38 vezes o segundo e 31 vezes o terceiro. Isso evidencia o impacto do efeito *ping-pong* desse modelo para este cenário, além de apresentar mais de três execuções (< 3) em quatro casos, destacando assim, as múltiplas trocas de célula durante o deslocamento.

No que se refere ao Cenário 2, a Figura 4 mostra que ele apresentou o maior número de falhas. Esse resultado deve-se à complexidade do deslocamento, já que o usuário precisa realizar dois *handovers* em um curto período. Para este cenário, são esperados 150 *handovers* da célula inicial para a alvo (legenda 1) e o no retorno para a célula inicial (legenda 2). Para o DDQN temos que os 150 primeiros *handovers* foram realizados com sucesso, contudo, a volta apresentou falha em 12% dos casos, fazendo com que o usuário não retornasse para a célula inicial, além da execução do terceiro *handover* em 6% dos casos, ou seja, após retornar para a célula inicial, ele realizou um novo *handover* para a célula vizinha. O GRU não executou o primeiro *handover* 7,3% das vezes e 19,3%

das vezes para o segundo, além de apresentar 9% de realizações em excesso.

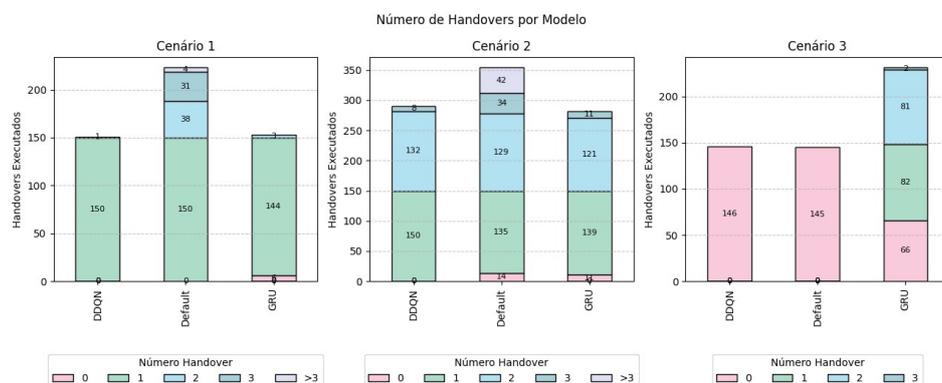


Figura 4. Número total de *handovers* executados nas simulações.

Por fim, ainda para o cenário 2, o modelo *default* apresentou o maior número de falhas, tal como a não realização em 10% dos casos. Além disso, ele evidencia o efeito *ping-pong*, onde o modelo executa *handovers* diversas vezes, chegando a realizar mais do que 4 *handovers* em 33% das vezes, resultando assim um total de 36,6% de falhas ao longo das simulações. Isso se deve ao fato dos tempos de disparo do *handover* para esse cenário estarem próximos, conforme mostrado na Figura 3.

Finalmente, no cenário 3, não se espera a realização de *handover*. Os modelos DDQN e *Default* exibiram comportamentos semelhantes para este cenário. O DDQN registrou uma taxa de erro de cerca de 2,67%, enquanto o *Default* apresentou 3,33%. O GRU foi o menos eficiente, pois não conseguiu detectar a saída do usuário da área de cobertura da célula atual, resultando em uma taxa de erro de 56%.

As consequências do tempo de disparo e quantidade de realizações de *handover* podem ser vistas diretamente nas métricas de QoS, tais como atraso e perda de pacotes. Sendo assim, essas métricas foram utilizadas para avaliação do desempenho da rede para cada modelo. Assim, como nas análises sobre o tempo de disparo, fez-se uso de valores ótimos para quantificar as perdas e ganhos. Para obtenção dos valores de referência (linhas tracejadas vermelhas) das Figuras 5 e 6, o usuário foi posicionado de maneira estática e a aplicação de envio de pacotes configurada da mesma maneira que a utilizada para os testes com mobilidade, totalizando 17334 pacotes com um atraso de 8ms.

O atraso médio final e a perda de pacotes podem ser vistos nas Figuras 5 e 6, respectivamente. Observa-se comportamento similar nos cenários 1 e 2. O atraso pode ser intensificado devido ao retardo ou antecipação do mecanismo para disparar o *handover*, resultando em um aumento da distância entre o usuário e a célula à qual ele está conectado. O *Default* apresentou o maior atraso em ambos os cenários, devido ao seu distanciamento em relação ao momento ideal de disparo do *handover*, com um aumento do atraso médio de 39% no cenário 1 e 36,38% no cenário 2. O mecanismo que emprega GRU apresentou um aumento do atraso médio de apenas 24% no cenário 1 e 31,8% no cenário 2. A proposta utilizando DDQN, por sua vez, obteve os melhores resultados devido à sua maior assertividade em relação ao tempo de disparo.

Em relação à perda de pacotes, considerando o aprimoramento da qualidade de sinal proporcionada pelas redes aéreas [Queiroz et al. 2023], um fator de elevado impacto

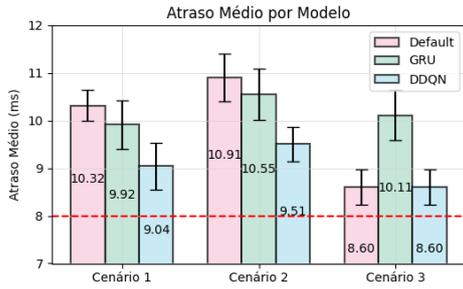


Figura 5. Atraso Médio.

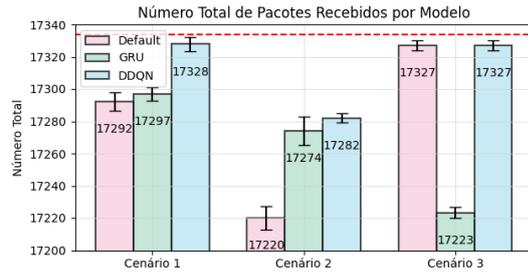


Figura 6. Pacotes recebidos.

seria a ocorrência de *handovers* desnecessários. No cenário 1, observa-se que o DDQN praticamente não apresentou perda de pacotes. Por outro lado, o *Default* e o GRU apresentaram algumas falhas de realização de *handovers*, resultando em uma perda de aproximadamente 600% e 517% a mais do que o DDQN, respectivamente. No cenário 2, o *Default* apresentou o maior número de falhas, resultando em uma perda de cerca de 120% a mais em comparação com o DDQN, que foi o modelo com menor perda de pacotes.

Nas Figuras 5 e 6, observa-se que os modelos *Default* e DDQN alcançaram resultados próximos no cenário 3 em relação ao atraso médio e uma similaridade em relação à quantidade de pacotes recebidos. Isso reflete o comportamento semelhante de ambos no número de *handovers* executados em relação a este cenário, por serem capazes de prever a não realização de *handovers*. Nesse cenário, destacam-se apenas as perdas e o atraso resultantes da mobilidade do usuário. Por outro lado, o GRU apresentou diversas falhas, realizando vários *handovers*, o que impactou negativamente nas métricas de QoS.

7. Conclusão

Este trabalho propôs um mecanismo de *handover* baseado em aprendizado por reforço profundo empregando o algoritmo DDQN. Este modelo se caracteriza por ser sensível ao contexto, determinando a ação a partir do estado do usuário. Para validar o modelo proposto, foi utilizado um ambiente de simulação no OMNeT++, onde extensões foram implementadas no *Simu5G* para acomodar as características de uma rede aérea.

Os resultados obtidos evidenciam que o melhor modelo para a realização do gerenciamento de *handover* para os cenários avaliados neste artigo é a solução baseada no DDQN. O modelo proposto apresentou a maior aproximação dos valores ideais para a realização de *handover* e obteve 74% de mais acertos quando comparado ao modelo *Default*, isto para os deslocamentos que realizaram 1 e 2 *handovers* (cenários 1 e 2). O DDQN apresentou uma redução de 95% nas falhas em comparação com o GRU, no cenário onde não há ocorrência de *handover* (cenário 3). Os resultados obtidos na avaliação da qualidade de serviço da rede evidenciam que para os cenários 1 e 2 o mecanismo *Default* apresentou o pior resultado em todas as métricas analisadas, consequência das falhas de *handover*. Para o cenário 3, o DDQN e o *default* obtiveram resultados similares. E, por fim, o DDQN foi o modelo que apresentou os melhores valores para as métricas de QoS em relação aos demais mecanismos, evidenciando a eficácia da utilização conjunta dos parâmetros de mobilidade e redes para a predição de *handover*.

Como trabalho futuro, a solução proposta pode ser melhorada movendo o moni-

toramento do contexto do usuário para o núcleo da rede 5G. Além disso, é fundamental realizar novas simulações, expandindo o ambiente com um maior número de UAV-BSs e UEs. Isso permitirá avaliar a escalabilidade e seus impactos em diferentes cenários. Por fim, é essencial realizar uma análise detalhada do consumo de energia das UAV-BSs nos cenários e modelos de aprendizado abordados neste artigo.

Referências

- Alsoliman, A., Rigoni, G., Levorato, M., Pinotti, C., Tippenhauer, N. O., and Conti, M. (2021). Cots drone detection using video streaming characteristics. In *Proceedings of the 22nd International Conference on Distributed Computing and Networking*.
- Aydin, Y., Kurt, G. K., Ozdemir, E., and Yanikomeroglu, H. (2021). Group handover for drone base stations. *IEEE Internet of Things Journal*, 8(18):13876–13887.
- Derhab, A., Cheikhrouhou, O., Allouch, A., Koubaa, A., Qureshi, B., Ferrag, M. A., Maglaras, L., and Khan, F. A. (2023). Internet of drones security: Taxonomies, open issues, and future directions. in *Vehicular Communications*.
- Gangula, R., Esrafilian, O., Gesbert, D., Roux, C., Kaltenberger, F., and Knopp, R. (2018). Flying rebots: First results on an autonomous uav-based lte relay using open airinterface. *IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications*.
- Hasselt, H. v., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press.
- Hu, H., Yang, L., and Wang, S. (2019). A trajectory prediction based intelligent handover control method in UAV cellular networks. *China Communications*, 16(1):1–14.
- Jang, Y., Raza, S. M., Kim, M., and Choo, H. (2022). Proactive handover decision for uavs with deep reinforcement learning. *Sensors*, 22(3).
- Lin, X. (2022). An overview of 5g advanced evolution in 3gpp release 18. *IEEE Communications Standards Magazine*, 6(3):77–83.
- Mishra, D. and Natalizio, E. (2020). A survey on cellular-connected uavs: Design challenges, enabling 5g/b5g innovations, and experimental advancements. *Computer Networks*.
- Muruganathan, S. D., Lin, X., Määtänen, H.-L., Sedin, J., Zou, Z., Hapsari, W. A., and Yasukawa, S. (2021). An overview of 3gpp release-15 study on enhanced lte support for connected drones. *IEEE Communications Standards Magazine*, 5(4):140–146.
- Queiroz, A., Barbosa, M. K., and Dias, K. (2023). Aero5gbs—deep learning-empowered ground users handover in aerial 5g and beyond systems. *IEEE Access*.
- Tafintsev, N., Chiumento, A., Vikhrova, O., Valkama, M., and Andreev, S. (2023). Utilization of uavs as flying base stations in urban environments. In *2023 15th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*.
- Yang, H., Hu, B., and Wang, L. (2017). A deep learning based handover mechanism for UAV networks. In *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*. IEEE.