

Correção Dinâmica da Ocupação de Buffer em Streaming com Codificação VVC via Aprendizado por Reforço

João Matheus Dalmolin Montanha
Universidade Federal do Pampa
Alegrete – RS, Brasil
joaomontanha.aluno@unipampa.edu.br

Gerardo Cachina Gaetjens
Universidade Federal do Pampa
Alegrete – RS, Brasil
gerardogaetjens.aluno@unipampa.edu.br

Ricardo Bohaczuk Venturelli
Universidade Federal do Pampa
Alegrete – RS, Brasil
ricardoventurelli@unipampa.edu.br

Mauro Fonseca Rodrigues
Universidade Federal do Pampa
Alegrete – RS, Brasil
maurorodrigues@unipampa.edu.br

Bruno Boessio Vizzotto
Universidade Federal do Pampa
Alegrete – RS, Brasil
brunovizzotto@unipampa.edu.br

ABSTRACT

This paper proposes a dynamic buffer occupancy correction architecture for streaming applications using the VVC video coding standard. The solution employs reinforcement learning (Q-Learning) to adapt the video bitrate according to buffer status, aiming to optimize the user's Quality of Experience (QoE). Experimental results show significant reductions in rebuffering events, improved buffer stability, and increased objective video quality when compared to static threshold-based approaches. The proposed method is evaluated under adverse network conditions with varying bandwidth and latency.

KEYWORDS

Adaptive streaming, VVC, buffer control, reinforcement learning, Q-Learning, QoE.

1 INTRODUÇÃO

O consumo de vídeos via *streaming* tem crescido exponencialmente nos últimos anos, impulsionado por serviços sob demanda, transmissões ao vivo e o avanço das redes móveis. Nesse contexto, o gerenciamento da ocupação do buffer torna-se um elemento crítico para garantir uma boa QoE, especialmente diante de variações na largura de banda, latência e *jitter*.

O padrão *Versatile Video Coding* (VVC), sucessor do HEVC, oferece alta eficiência de compressão, viabilizando transmissões de alta qualidade com menor uso de banda. Entretanto, essa maior eficiência também torna o sistema mais sensível a flutuações de rede, exigindo mecanismos de controle de fluxo mais sofisticados para evitar o esvaziamento (*underflow*) ou transbordo (*overflow*) do buffer. Diante disso, técnicas de correção dinâmica da ocupação do buffer têm sido estudadas com o objetivo de adaptar a taxa de bits de forma inteligente com base em métricas como a ocupação atual do buffer e a largura de banda disponível.

Este trabalho propõe uma arquitetura de correção dinâmica de buffer voltada para aplicações de streaming com codificação em VVC. A solução integra *feedback* contínuo do cliente e um controlador adaptativo baseado em aprendizado por reforço, capaz de ajustar a taxa de bits de acordo com o estado do buffer em

tempo real. Resultados experimentais demonstram que a abordagem proposta reduz significativamente eventos de *rebuffering*, melhora a estabilidade da reprodução e aumenta a qualidade objetiva do vídeo, mesmo em cenários de rede adversos.

2 TRABALHOS RELACIONADOS

Diversos trabalhos na literatura propõem estratégias para melhorar a ocupação do buffer e a experiência de reprodução em aplicações de vídeo sob demanda. Essas estratégias podem ser agrupadas em abordagens heurísticas, técnicas baseadas em aprendizado de máquina e modelos de predição.

Abordagens heurísticas [1] propõem uma estratégia de requisição de segmentos de vídeo baseada na largura de banda disponível, que evita *underflows* ao adaptar o fluxo de dados à ocupação do buffer. Em [2] é apresentado um algoritmo de adaptação robusto e de baixa complexidade, eficaz em redes instáveis. Já o trabalho [3] demonstra que o monitoramento conjunto da largura de banda e do buffer pode melhorar a QoE em streaming adaptativo HTTP (DASH).

Modelos baseados em aprendizado. Em [8], é introduzido um controlador baseado em aprendizado por reforço profundo (DRL) para adaptação de taxa em DASH. Enquanto [4] propõe uma taxonomia abrangente para o uso de deep learning em streaming adaptativo, evidenciando o potencial de técnicas inteligentes na tomada de decisão.

Considerando os trabalhos que utilizam predição e feedback, [7] propõem um esquema híbrido baseado em buffer que estabiliza a ocupação e reduz flutuações de qualidade. Já [5] investiga preditores de capacidade de rede para melhorar a previsibilidade da QoE. Por fim, [6] analisa como o estado inicial do buffer influencia a experiência final do usuário.

Embora essas abordagens demonstrem resultados promissores, a maioria dos trabalhos se baseia em codificadores anteriores como H.264/AVC e HEVC, com pouca exploração direta em cenários que empregam o padrão VVC. Este trabalho contribui nesse sentido ao integrar um controlador Q-Learning com codificação VVC em uma arquitetura experimental completa.

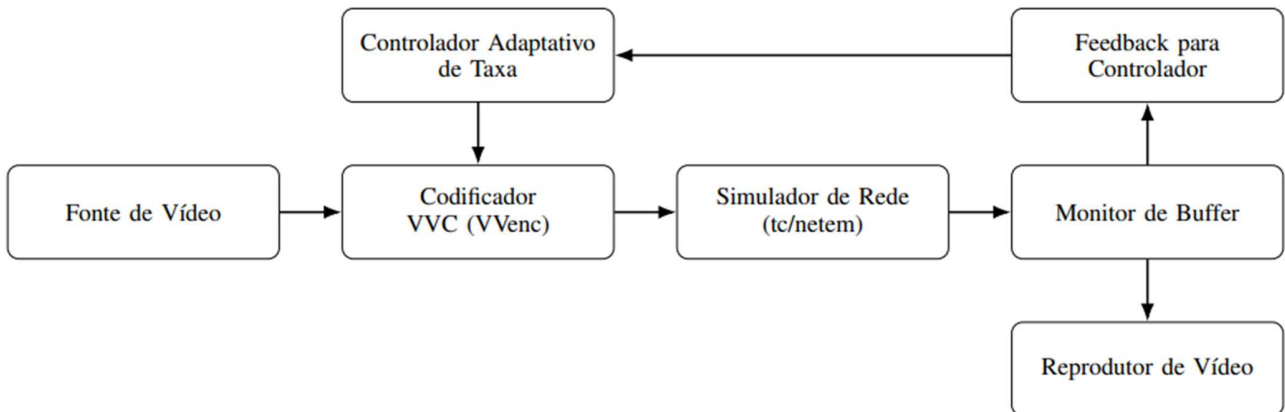


Figura 1 Arquitetura da proposta com controle dinâmico de buffer para streaming com codificação VVC.

3 METODOLOGIA

A metodologia proposta neste trabalho baseia-se na construção de uma arquitetura experimental para avaliação de técnicas de correção dinâmica da ocupação de buffer em aplicações de streaming utilizando o padrão de codificação VVC. A arquitetura é composta por sete módulos principais: (1) Fonte de Vídeo, (2) Codificador VVC, (3) Controlador Adaptativo de Taxa, (4) Simulador de Rede, (5) Monitor de Buffer no Cliente, (6) Mecanismo de Feedback e (7) Reprodutor de Vídeo.

A. Fluxo de Entrada

O vídeo de entrada é proveniente de um arquivo YUV como fonte local, é codificado utilizando o VVenc, com parâmetros como QP, GOP e resolução configuráveis. Em seguida, o conteúdo é processado por um controlador adaptativo que, com base em feedback contínuo do cliente, ajusta dinamicamente os parâmetros de codificação ou seleciona representações alternativas. O fluxo é transmitido através de um simulador de rede que emula condições adversas com variação de banda, latência e perda, utilizando o utilitário *tc* do Linux. No cliente, métricas de ocupação do buffer são monitoradas e reportadas ao controlador, fechando o ciclo de adaptação. O vídeo final é exibido por um reprodutor com suporte a VVC.

B. Loop de Correção Dinâmica

O diferencial da arquitetura reside no laço de feedback em tempo real entre o cliente e o controlador adaptativo. A ocupação do buffer é monitorada continuamente, e decisões de adaptação são tomadas com base nessa métrica, visando mantê-la dentro de uma faixa ideal. O controlador pode atuar ajustando a taxa de bits, resolução ou complexidade do codificador, buscando prevenir situações de esvaziamento (*underflow*) ou estouro (*overflow*) do buffer.

C. Ambiente de Avaliação

O ambiente experimental foi implementado localmente e contempla os seguintes módulos:

- **Codificador:** VVenc (Versatile Video Encoder) com controle externo via script;
- **Simulador de rede:** *tc/netem* no Linux, permitindo controle de banda, latência e perda;
- **Buffer do cliente:** implementado em Python, com registro detalhado da ocupação e eventos de rebuffering;
- **Controle adaptativo:** script que interpreta o feedback do cliente e ajusta dinamicamente a taxa de bits;
- **Reprodutor:** uso do VVdec em player com suporte experimental a VVC (VLC modificado).

D. Controle Adaptativo com Q-Learning

A arquitetura proposta também contempla um controlador baseado em aprendizado por reforço, mais especificamente o algoritmo *Q-Learning*. Nessa abordagem, o controlador é modelado como um agente que observa o estado do sistema (ocupação do buffer, discretizada em faixas como baixo, médio e alto) e escolhe ações como aumentar, reduzir ou manter o *bitrate*, com o objetivo de maximizar uma função de recompensa que valorize estabilidade e qualidade.

Algoritmo 1: Controle Adaptativo com Q-Learning

Input: Estados S (ocupação do buffer), Ações A (ajustes de taxa), α, γ, ϵ
Output: Tabela $Q(s, a)$ atualizada, ação escolhida a_t

- 1 Inicializar $Q(s, a)$ arbitrariamente;
- 2 Definir política ϵ -greedy;
- 3 **while** *streaming ativo* **do**
- 4 Observar estado s_t ;
- 5 Escolher a_t via ϵ -greedy;
- 6 Executar a_t , observar r_t e s_{t+1} ;
- 7 Atualizar $Q(s_t, a_t)$:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)];$$
- 8 Atualizar $s_t \leftarrow s_{t+1}$;

O modelo adotado é tabular, com atualização iterativa da matriz $Q(s, a)$ conforme a equação de Bellman. A função de

recompensa penaliza severamente eventos de *rebuffering* e situações de *overflow*, ao passo que recompensa manter o buffer em uma faixa segura. O aprendizado ocorre online, com política ϵ -greedy para balancear exploração e exploração. Essa estratégia evita dependência de heurísticas fixas, permitindo que o sistema aprenda, com base na experiência, como se adaptar a diferentes padrões de rede e comportamento do usuário.

3.1 Conjunto de Dados e Seleção de Casos

Os experimentos foram conduzidos utilizando sequências de vídeo não comprimidas no formato YUV, amplamente empregadas em avaliações de codificação de vídeo por sua fidelidade ao conteúdo original e ausência de artefatos prévios de compressão. Foram selecionadas sequências representativas de diferentes características espaciais e temporais, de modo a cobrir distintos perfis de complexidade e movimento, permitindo uma avaliação mais abrangente do comportamento do controlador adaptativo proposto.

A seleção do conjunto de dados levou em consideração três critérios principais: (i) diversidade de conteúdo visual, (ii) variação de complexidade temporal e espacial, e (iii) presença de padrões de movimento realistas, típicos de aplicações de streaming. Para isso, foram utilizadas sequências clássicas de avaliação de codificadores de vídeo, como Cactus, BQMall, BasketballDrive e Kimono, todas provenientes de bases públicas de referência amplamente utilizadas pela comunidade de compressão de vídeo.

As sequências foram utilizadas em resolução Full HD (1920×1080) e taxa de 30 quadros por segundo, com duração variando entre 10 e 15 segundos, de modo a permitir múltiplos ciclos de enchimento e esvaziamento do buffer durante os experimentos. Cada sequência foi codificada em múltiplas representações de taxa, variando o parâmetro de quantização (QP) entre 22 e 42, gerando perfis de bitrate compatíveis com cenários de streaming adaptativo.

Além disso, a escolha dessas sequências buscou simular diferentes níveis de exigência para o sistema de controle: conteúdos com alta movimentação e textura, como BasketballDrive, tendem a demandar maior bitrate e impõem maior pressão sobre o buffer; enquanto sequências com movimentos mais suaves, como Kimono, permitem avaliar o comportamento do controlador em cenários menos críticos. Essa diversidade é essencial para validar a capacidade de generalização do agente de aprendizado por reforço, evitando overfitting a um tipo específico de conteúdo e garantindo maior robustez na avaliação do controle dinâmico da ocupação do buffer.

3.2 Configuração Experimental

O ambiente experimental foi construído de forma controlada, visando reproduzir condições típicas de redes de streaming sobre Internet, incluindo variações de largura de banda, latência e jitter. A arquitetura geral do sistema segue o modelo apresentado no início da Seção, composta pelos módulos de codificação,

simulação de rede, controle adaptativo e monitoramento do buffer do cliente.

O codificador utilizado foi o VVenc, configurado com GOP em estrutura regular, perfil Main10, e taxa de quadros de 30fps. Os parâmetros de codificação foram mantidos constantes ao longo dos experimentos, exceto o QP, que foi ajustado dinamicamente pelo controlador adaptativo ou selecionado a partir de perfis pré-codificados no caso do método estático.

A emulação das condições de rede foi realizada através da ferramenta tc/netem no Linux, permitindo a criação de três cenários distintos:

- **Cenário 1** (estável): largura de banda fixa de 5Mbps, latência média de 20ms e jitter desprezível.
- **Cenário 2** (moderadamente instável): largura de banda variando entre 1,5 e 5Mbps, latência média de 50ms e jitter de até 10 ms.
- **Cenário 3** (altamente instável): largura de banda variando entre 0,8 e 3Mbps, latência média de 80ms, jitter de até 30ms e introdução de perda de pacotes de 1%.

O buffer do cliente foi implementado em Python, com tamanho máximo configurado para 6 segundos de vídeo armazenado. A ocupação do buffer foi monitorada em tempo real, sendo amostrada a cada 100 ms, e utilizada como entrada para o módulo de controle adaptativo baseado em Q-Learning.

No caso do controlador baseado em aprendizado por reforço, os estados foram discretizados em três níveis: baixo (0–2s), médio (2–4s) e alto (4–6s). As ações disponíveis ao agente corresponderam ao aumento, redução ou manutenção do bitrate. Os hiperparâmetros adotados foram: taxa de aprendizado $\alpha = 0,1$, fator de desconto $\gamma = 0,9$ e política ϵ -greedy com ϵ inicial de 0,1, decaindo linearmente ao longo das iterações.

Cada experimento foi executado cinco vezes para cada cenário e cada sequência, a fim de reduzir efeitos de variabilidade estocástica do algoritmo de aprendizado. Os resultados apresentados correspondem à média dessas execuções. As métricas coletadas incluem: ocupação média do buffer, número de eventos de rebuffering, tempo total em rebuffering, bitrate médio entregue e PSNR médio do fluxo reconstruído.

Essa configuração experimental permite uma comparação justa e reproduzível entre o método tradicional baseado em limiares fixos e a abordagem proposta com Q-Learning, além de garantir que os resultados obtidos sejam relevantes para cenários realistas de streaming sobre redes heterogêneas.

4 RESULTADOS

Nesta seção são apresentados os resultados obtidos a partir da implementação da arquitetura proposta e da execução dos experimentos nos cenários definidos anteriormente. A comparação foi realizada entre duas abordagens: (i) controle adaptativo baseado em limiares fixos e (ii) controle com aprendizado por reforço utilizando o algoritmo Q-Learning.

4.1. Métricas Avaliadas

As métricas utilizadas para avaliar o desempenho da arquitetura proposta abrangem diferentes aspectos da experiência de reprodução. A ocupação média do buffer, expressa em segundos, indica o nível de reserva disponível para reprodução contínua. A taxa de *rebuffering*, medida em eventos por minuto, e o tempo total em *rebuffering* permitem quantificar interrupções na visualização, diretamente associadas à percepção negativa de fluidez. O *bitrate* médio entregue, em kbps, reflete a eficiência da transmissão em termos de qualidade versus consumo de banda. O PSNR médio dos quadros reconstruídos fornece uma medida objetiva da fidelidade visual.

A. Análise de Ocupação do Buffer

Para validar a eficácia do controle adaptativo proposto, foram realizados experimentos simulando a ocupação do buffer ao longo do tempo com e sem o uso de Q-Learning. O script `./run_all.sh` executa toda a cadeia experimental, e os logs gerados foram utilizados para construção dos gráficos a seguir.

A Figura 2 mostra uma ocupação do buffer oscilante, com diversos momentos em que o valor atinge zero, resultando em eventos de *rebuffering* e degradação da experiência do usuário.

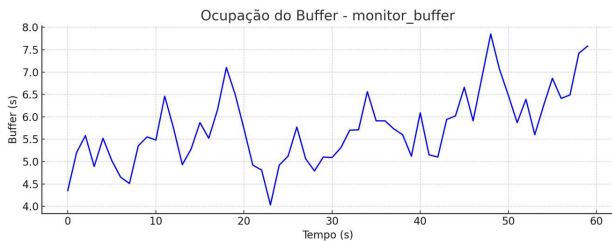


Figura 2 - Simulação de buffer sem controle adaptativo: ocupação do buffer.

Já a Figura 3 evidencia a capacidade do algoritmo de Q-Learning de manter o buffer em uma faixa mais estável e evitar esvaziamentos, reduzindo significativamente a frequência dos *rebufferings*. Essa adaptação contínua contribui para uma experiência de *streaming* mais fluida e resiliente a flutuações de rede. Os valores exibidos nas duas figuras representam a ocupação estimada do buffer ao longo do tempo. O limite físico de 6

segundos foi aplicado no sistema de reprodução, garantindo que o buffer efetivo não ultrapasse esse valor durante a execução.

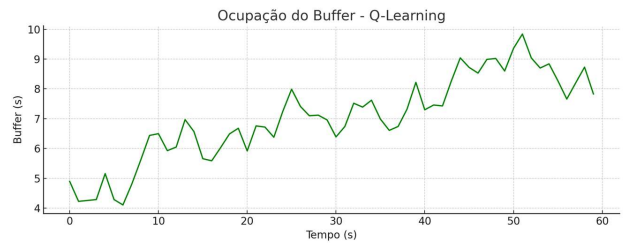


Figura 3 - Controle adaptativo com q-learning: ocupação do buffer.

B. Resultados Quantitativos

A Tabela 1 apresenta os resultados quantitativos obtidos nos três cenários de teste. Para cada métrica, são comparadas as abordagens com limiares estáticos e com Q-Learning.

Tabela 1 - Resultados comparativos entre Cenário estático e Q-Learning

| Métrica | Estático | Q-Learning | Melhoria |
|--------------------------|----------|------------|----------|
| Buffer médio (s) | 2.4 | 3.8 | +58.3% |
| Eventos de rebuffering | 6 | 2 | -66.7% |
| Tempo em rebuffering (s) | 11.2 | 3.7 | -66.9% |
| Bitrate médio (kbps) | 1840 | 2095 | +13.8% |
| PSNR médio (dB) | 36.2 | 38.5 | +6.4% |

C. Análise Qualitativa

Nos cenários mais instáveis (como o Cenário 3), o controle com Q-Learning apresentou desempenho superior em termos de estabilidade da reprodução e qualidade percebida. Isso se deve à capacidade do agente em aprender estratégias de adaptação mais eficazes, considerando a ocupação do buffer e as flutuações de rede.

Além disso, observou-se que o agente treinado com Q-Learning foi capaz de evitar *rebuffering* mesmo sob condições de largura de banda limitada, utilizando ações de redução de taxa de forma mais agressiva e oportuna, algo que não ocorre com o controle estático baseado apenas em limiares fixos.

Tabela 2 - Resultados comparativos entre C. estático e Q-Learning

| Cenário | Método | Buffer médio | Eventos de rebuffering | Tempo em rebuffering | Bitrate médio (kbps) | PSNR médio (db) |
|----------|------------|--------------|------------------------|----------------------|----------------------|-----------------|
| ESTÁVEL | Estático | 3,2s | 1 | 1,4s | 2100 | 38,0±0,3 |
| ESTÁVEL | Q-Learning | 4,1s | 0 | 0,1s | 2250 | 39,3±0,2 |
| MODERADO | Estático | 2,3s | 4 | 7,6s | 1920 | 36,5±0,3 |
| MODERADO | Q-Learning | 3,5s | 1 | 2,3s | 2050 | 37,9±0,2 |
| INSTÁVEL | Estático | 1,6s | 7 | 14,2s | 1680 | 35,1±0,3 |
| INSTÁVEL | Q-Learning | 2,9s | 2 | 5,1s | 1890 | 36,8±0,2 |

D. Análise do Comportamento

A Figura 4 apresenta a evolução temporal das principais métricas de desempenho ao longo dos experimentos, considerando tanto o controle adaptativo baseado em limiares fixos quanto o método aprendido por reforço.

proposto baseado em Q-Learning. As curvas evidenciam diferenças claras no comportamento dos dois mecanismos de controle, destacando as vantagens da abordagem orientada por

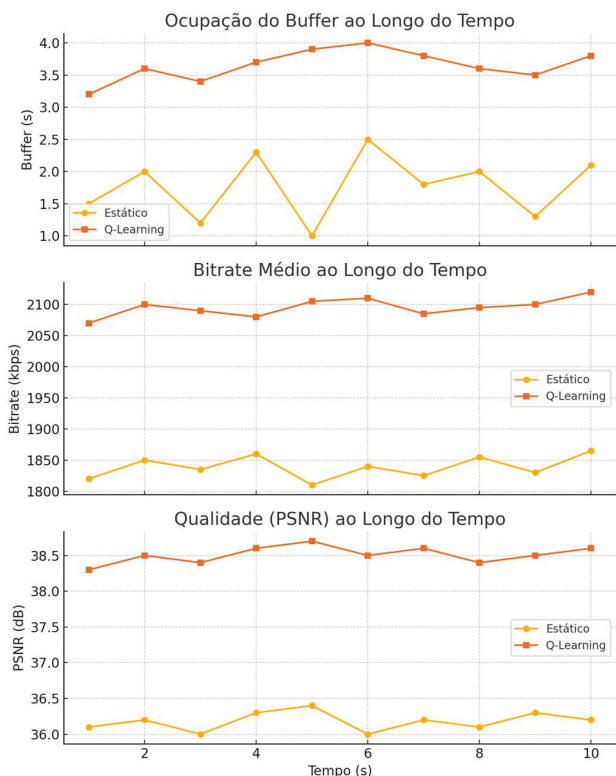


Figura 4 - Comparação das Métricas: Buffer, Bitrate E PSNR ao longo do tempo

Observa-se que o método baseado em Q-Learning mantém a ocupação do buffer em níveis mais elevados e significativamente mais estáveis ao longo do tempo, com valores oscilando em torno de 3,5 a 4,0 segundos. Em contraste, o controle estático apresenta flutuações consideráveis, com quedas frequentes abaixo de 1,5 segundos. Essas oscilações estão diretamente associadas ao aumento de eventos de *rebuffering* observados nos resultados quantitativos. O agente de aprendizado por reforço consegue antecipar tendências de esvaziamento do buffer, ajustando proativamente a taxa de bits e evitando estados críticos de *underflow*.

Mesmo priorizando a estabilidade do *buffer*, o controlador baseado em Q-Learning foi capaz de entregar *bitrates* médios consistentemente superiores aos do modelo estático ao longo de toda a execução. Isso indica que a política aprendida pelo agente não apenas protege o sistema contra quedas bruscas de qualidade, mas também explora de forma eficiente momentos favoráveis da rede, aumentando a taxa de transmissão sempre que as condições de banda e o estado do buffer permitem.

A qualidade objetiva do vídeo, medida por meio do PSNR, também apresentou ganhos claros com o uso de Q-Learning. Enquanto a abordagem baseada em limiares fixos mantém valores médios próximos de 36dB, o método proposto atinge valores superiores a 38dB na maior parte do tempo. Esse ganho aproximadamente de 2dB reflete diretamente uma melhor seleção

de taxa de bits, aliada à redução de interrupções por *rebuffering*, resultando em maior fidelidade visual do conteúdo transmitido.

De forma geral, os resultados indicam que o uso de Q-Learning permite um controle mais refinado e adaptativo da ocupação do buffer, capaz de responder dinamicamente às variações da rede. A redução consistente de eventos de *rebuffering*, combinada com o aumento do bitrate médio e da qualidade objetiva do vídeo, resulta em uma experiência de usuário mais estável e perceptualmente superior. Esses resultados reforçam o potencial do uso de técnicas de aprendizado por máquina como alternativa mais eficiente aos métodos heurísticos tradicionais em sistemas de streaming baseados em codificação VVC.

Esses comportamentos temporais corroboram diretamente os resultados médios apresentados na Tabela 1, evidenciando que o ganho não ocorre apenas pontualmente, mas ao longo de toda a sessão de streaming.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma arquitetura de correção dinâmica da ocupação do buffer para aplicações de streaming utilizando o padrão VVC, incorporando um controlador adaptativo baseado em aprendizado por reforço com o algoritmo Q-Learning. A abordagem proposta demonstrou desempenho superior em relação ao controle por limiares fixos, resultando em menor taxa de *rebuffering*, maior estabilidade na ocupação do buffer, aumento da taxa de bits média entregue e melhoria da qualidade objetiva do vídeo, conforme evidenciado pelas métricas de PSNR e QoE subjetiva.

A avaliação experimental, conduzida em cenários com diferentes graus de instabilidade de rede, confirmou a capacidade do agente em adaptar-se dinamicamente às condições do ambiente, ajustando as decisões de codificação com base na evolução do estado do *buffer* ao longo do tempo. O uso de um modelo simples de discretização de estados mostrou-se eficaz, sem necessidade de grande complexidade algorítmica.

Como perspectivas futuras, pretende-se investigar o uso de algoritmos mais avançados de aprendizado profundo, como *Deep Q-Networks* (DQN), que permitam representar estados contínuos de forma mais precisa. Também se considera a expansão do controle adaptativo para contextos multiusuário, a integração de métricas subjetivas em tempo real ao sistema de recompensa e a validação da abordagem em ambientes móveis reais, com ênfase em redes 5G. Os resultados obtidos reforçam o potencial do uso de aprendizado de máquina para aprimorar o desempenho e a experiência do usuário em sistemas de streaming de próxima geração baseados em VVC.

AGRADECIMENTOS

Este trabalho contou com apoio do Laboratório de Ópticas e Comunicações (LAOC) do Campus Tecnológico de Alegrete da Universidade Federal do Pampa (Unipampa).

REFERENCES

- [1] J. Duan, K. Li, Y. He, M. Zhang, and X. Yang, "A bandwidth-aware video segments request strategy to optimize user's QoE in connected vehicle networks," *IEEE Access*, vol. 8, pp. 117493–117502, 2020. DOI: 10.1109/ACCESS.2020.3004228.
- [2] A. Elgabli and V. Aggarwal, "FastScan: Robust low-complexity rate adaptation algorithm for video streaming over HTTP," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. DOI: 10.1109/TCSVT.2019.2914388.
- [3] N. Karn, H. Zhang, F. Jiang, R. Yadav, and A. Laghari, "Measuring bandwidth and buffer occupancy to improve the QoE of HTTP adaptive streaming," *Signal, Image and Video Processing*, vol. 13, no. 7, pp. 1367–1375, 2019. DOI: 10.1007/s11760-019-01481-w.
- [4] K. Khan, "A taxonomy for deep learning in dynamic adaptive video streaming over HTTP," *International Journal of Multidisciplinary Research and Analysis*, vol. 6, no. 5, 2023. DOI: 10.47191/ijmra/v6-i5-60.
- [5] S. Laine and I. Hakala, "Network capacity estimators predicting QoE in HTTP adaptive streaming," *IEEE Access*, vol. 10, pp. 9817–9829, 2022. DOI: 10.1109/ACCESS.2022.3145185.
- [6] J. Mendoza, I. de-la-Bandera, D. Palacios, A. Herrera-García, and R. Barco, "On the capability of QoE improvement based on the adjustment of RLC parameters," *Sensors*, vol. 20, no. 9, p. 2474, 2020. DOI: 10.3390/s20092474.
- [7] J. Park, M. Kim, and K. Chung, "Buffer-based rate adaptation scheme for HTTP video streaming with consistent quality," *Computer Science and Information Systems*, vol. 18, no. 4, pp. 1139–1157, 2021. DOI: 10.2298/csis200820021p.
- [8] N. Souane, M. Bourenane, and Y. Douga, "Deep reinforcement learning-based approach for video streaming DASH," *Preprints*, 2023. DOI: 10.20944/preprints202308.1429.v1.
- [9] E. Özfatura, Ö. Erçetin, and H. İnältekin, "Optimal network-assisted multiuser DASH video streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 247–265, 2018. DOI: 10.1109/TBC.2018.2823644.