Avaliação de Estimadores de Largura de Banda e Impactos em Aplicações de Vídeo 360º em Tempo Real

Públio Elon Correa da Silva[1] **, Fábio Luciano Verdi**[1]

[1] Universidade Federal de São Carlos (UFSCar)
Sorocaba, SP - Brasil

publio@estudante.ufscar.br, verdi@ufscar.br

Resumo. *Streaming de vídeo 360° em tempo real exige baixa latência e alta qualidade sob oscilações de rede. Sem estimativa de largura de banda, o codificador pode exceder a capacidade do enlace, causando filas e perdas, ou subutilizá-lo. Neste trabalho, avaliamos um controle adaptativo de taxa por chunk guiado por estimativas de largura de banda e, em paralelo, uma codificação por tiles guiada por saliência, que redistribui o bitrate entre regiões do quadro sem alterar a taxa global. Comparamos, sob o mesmo protocolo experimental, estimadores externos ao pipeline de compressão baseados em TCP_INFO e em goodput medido no receptor, além do Google Congestion Control (GCC) com Transport-Wide Congestion Control (TWCC) e de uma taxa fixa. Os resultados mostram que os estimadores preservam a qualidade em baixa largura de banda e, em alta largura de banda, contêm o crescimento do chunk mantendo qualidade semelhante. No melhor caso com saliência, o TCP_INFO elevou o Peak Signal-to-Noise Ratio (PSNR) em 1,25 dB com aumento negligenciável no tamanho do arquivo.*

Abstract. *Real-time 360° video streaming requires low latency and high visual quality under network fluctuations. Without bandwidth estimation, the encoder may exceed link capacity, causing queue buildup and losses, or underutilize it. This work evaluates adaptive per-chunk rate control driven by bandwidth estimates and, in parallel, a saliency-guided tile-based encoding scheme that redistributes bitrate across frame regions without changing the overall rate. Under a common experimental protocol, we compare estimators external to the compression pipeline based on TCP_INFO and receiver-measured goodput, alongside Google Congestion Control (GCC) with Transport-Wide Congestion Control (TWCC) and a fixed-rate configuration. Results show that the estimators preserve visual quality under low bandwidth and, under high bandwidth, curb chunk size growth while maintaining similar quality. In the best saliency case, TCP_INFO increased Peak Signal-to-Noise Ratio (PSNR) by 1.25 dB with negligible file-size increase.*

1. Introdução

A evolução das redes rumo ao 6G amplia a taxa de transmissão, reduz a latência e melhora a confiabilidade, enquanto a integração entre redes públicas e privadas viabiliza ambientes híbridos com maior controle de recursos [Jiang et al. 2021]. Nesse cenário, o crescimento do consumo de vídeo 4K/8K e de formatos imersivos em 360° exige dezenas a centenas

de megabits por segundo e torna o desempenho sensível a atrasos, de modo que abordagens tradicionais têm dificuldade em manter qualidade estável sob flutuações de capacidade causadas por congestionamento, compartilhamento de enlaces ou mobilidade [Khan et al. 2022].

Sistemas de comunicação em tempo real exigem adaptação contínua da taxa às condições da rede, mas o transmissor observa apenas sinais indiretos de realimentação como atraso, perdas e estimativas de largura de banda, o que leva controles de congestionamento e estimadores a ajustar a transmissão de forma reativa ou preditiva. Quando esse acompanhamento falha ou quando se opera em taxa fixa, a codificação pode exceder a capacidade e gerar filas, latência e perdas, ou ficar aquém do possível e desperdiçar banda; por isso, a estimação de largura de banda integrada ao controle de taxa ajuda a equilibrar eficiência e estabilidade e, combinada à codificação guiada por saliência e regiões de interesse, prioriza áreas visualmente relevantes e melhora a eficiência perceptual.

Este trabalho investiga o uso de estimadores de largura de banda para adaptar a taxa de codificação em *streaming* de vídeo 360° sob competição por capacidade no link de acesso do usuário. Em sistemas convencionais, o codificador opera sem comunicação explícita com a rede e não recebe informações sobre banda disponível no link; na prática, essa disponibilidade é condicionada por tráfego de fundo concorrente e pode variar ao longo do tempo. Nesse contexto, o objetivo é ajustar a taxa por *chunk* de forma compatível com a capacidade efetivamente disponível, evitando tanto congestionamento quanto subutilização.

A literatura aponta que, sob tráfego concorrente e variação de capacidade no link, a adaptação da taxa de codificação e da alocação por *tiles* contribui para preservar a qualidade de experiência (QoE). Motivado por esse cenário, este artigo propõe dois métodos de estimação de largura de banda para definir a taxa por *chunk*. No primeiro, uma conexão TCP auxiliar opera em paralelo, e o emissor utiliza métricas obtidas via TCP_INFO como referência de taxa. No segundo, o receptor calcula o *goodput* com base nos bytes de vídeo efetivamente entregues em cada intervalo e realimenta essa estimativa ao emissor para ajustar o *chunk* seguinte. Por fim, a taxa selecionada por cada método é combinada com uma codificação hierárquica guiada por saliência, que redistribui o bitrate no quadro sem alterar o controle de congestionamento.

O restante do artigo está organizado da seguinte forma: a Seção 2 discute os trabalhos relacionados; a Seção 3 descreve a codificação guiada por saliência; a Seção 4 apresenta os métodos de controle e estimação avaliados; a Seção 5 detalha a metodologia experimental; a Seção 6 analisa os resultados; e a Seção 7 conclui o trabalho.

2. Trabalhos Relacionados

Diversos trabalhos investigam a estimação de largura de banda como base do controle adaptativo em comunicações em tempo real. Em cenários com tráfego altamente variável e múltiplos gargalos, estimadores tradicionais tendem a instabilizar, causando oscilações de taxa e degradação da qualidade percebida [Li et al. 2014]. Esse quadro evidencia limitações de abordagens baseadas em modelos estáticos ou medições pontuais, motivando mecanismos mais responsivos e robustos.

O trabalho de Carlucci et al. [Carlucci et al. 2016] introduz o Google Congestion Control (GCC) como um mecanismo de controle adaptativo voltado a aplicações intera-

tivas em tempo real. O GCC estima a largura de banda disponível a partir de métricas de atraso e feedbacks de transporte, ajustando dinamicamente a taxa de envio e antecipando congestionamentos. Essa abordagem, hoje amplamente adotada em sistemas WebRTC, tornou-se uma referência para o controle de congestionamento em comunicações em tempo real.

Outros trabalhos avançaram na integração entre decisões de codificação e comportamento do transporte. Zhang *et al.* exploraram sinais de atraso e perda para ajustar parâmetros de codificação em tempo real [Zhang et al. 2019], enquanto o Salsify propôs uma arquitetura co-desenhada de transporte e codec para adaptar dinamicamente o bitrate, obtendo ganhos de latência e qualidade percebida [Fouladi et al. 2018]. Ainda assim, essas abordagens tipicamente assumem fluxos homogêneos e não tratam explicitamente a relevância visual espacial do conteúdo.

Em paralelo, estudos voltados à qualidade de experiência enfatizam que decisões de alocação devem considerar não apenas a taxa disponível, mas também a relevância perceptual do conteúdo transmitido [Zhang et al. 2023]. Nesse contexto, o presente trabalho se insere ao combinar mecanismos clássicos de estimação de largura de banda com estratégias de codificação orientadas por saliência, explorando a sinergia entre controle de rede e percepção visual para otimizar a transmissão de vídeo 360°.

Em adição, diversos trabalhos investigaram estratégias específicas para a transmissão eficiente de vídeo 360°, explorando principalmente a predição de viewport e a seleção adaptativa de *tiles*. Trabalhos como o *Sinusoidal Viewport Prediction* (SVP) modelam o movimento do usuário a partir de padrões periódicos de rotação da cabeça, permitindo antecipar o campo de visão com baixo custo computacional [Jiang et al. 2020]. De forma complementar, sistemas como o 360ProbDASH incorporam explicitamente a incerteza da predição ao distribuir a qualidade entre *tiles* de acordo com probabilidades de visualização, demonstrando ganhos consistentes de qualidade percebida em comparação à transmissão uniforme [Xie et al. 2017].

Outros estudos avançaram ao formular o problema de adaptação espacial como uma otimização explícita, buscando maximizar a qualidade esperada do viewport sob restrições de taxa [Nguyen et al. 2019]. Avaliações comparativas indicam ainda que decisões relacionadas à seleção de tiles, como o uso de margens espaciais, degradação progressiva e integração de informações de saliência, têm impacto direto na estabilidade visual e na experiência do usuário [Nguyen et al. 2020]. Apesar dos ganhos observados, essas abordagens geralmente assumem condições de rede estáticas ou perfis de largura de banda previamente definidos, limitando sua capacidade de resposta a variações rápidas do enlace.

3. Codificação guiada por saliência

A codificação guiada por saliência parte do princípio de que, em vídeos 360°, a atenção do usuário é intrinsecamente não uniforme no quadro. Assim, regiões de interesse (ROI), associadas ao campo de visão e a elementos visualmente relevantes, dominam a percepção de qualidade, enquanto áreas periféricas têm menor impacto perceptual. Explorar essa assimetria pode aumentar a eficiência do uso da largura de banda em *streaming* imersivo [Wang et al. 2022].

Neste trabalho, a saliência é empregada apenas como mecanismo auxiliar de

alocação espacial de bits, para avaliar como decisões de taxa definidas por estimadores de largura de banda interagem com uma codificação não uniforme. O modelo de predição de saliência não constitui contribuição deste estudo e é usado apenas para gerar mapas normalizados de relevância visual por quadro, representando a probabilidade de atenção do usuário em um cenário de visualização com óculos de realidade virtual.

A partir dos mapas de saliência, o quadro equiretangular é particionado em *tiles* com pesos hierárquicos de importância perceptual, definindo três níveis de prioridade: regiões mais salientes são codificadas com maior qualidade, regiões intermediárias com qualidade moderada e regiões menos salientes com maior compressão. Essa estratégia redistribui espacialmente o bitrate sem alterar a taxa global definida pelo mecanismo de controle de taxa, atuando apenas na alocação interna do codificador, conforme ilustrado na Figura 1.

Figura 1. Codificação guiada por saliência: os quadros possuem forma equiretangular e são segmentados em *tiles*, onde tiles em vermelho indicam regiões codificadas com maior qualidade, tiles em verde representam qualidade intermediária e tiles em preto correspondem a regiões submetidas a maior compressão, conforme a relevância perceptual estimada.

A codificação guiada por saliência é adotada para analisar seu papel complementar aos mecanismos de estimação de largura de banda: sob restrição de capacidade, ela prioriza regiões visualmente relevantes, preservando a qualidade percebida e deslocando a degradação para áreas de menor impacto perceptual. Do ponto de vista do sistema, a saliência não modifica o estimador nem o ciclo de controle de taxa, atuando apenas como política interna de redistribuição hierárquica dos bits gerados pelo codificador.

4. Algoritmos de Controle de Congestão e Estimação de Largura de Banda

Nesta seção descrevem-se os métodos de controle de congestionamento e de estimação de largura de banda avaliados neste trabalho, bem como as equações usadas para estimar a capacidade do link e definir a taxa de codificação do vídeo.

4.1. Google Congestion Control (GCC)

O Google Congestion Control (GCC) é um mecanismo de controle de congestionamento amplamente adotado em aplicações interativas em tempo real baseadas em WebRTC. Seu funcionamento é guiado por realimentação do receptor via Transport-Wide Congestion Control (TWCC), que reporta os tempos de chegada dos pacotes. A partir desses tempos, o emissor infere variações de atraso ao longo do tempo, associadas à formação ou dissipação de filas no enlace, e ajusta dinamicamente a taxa de envio.

No GCC, a estimativa de largura de banda é ajustada com base nas tendências desse atraso observadas via TWCC:

$$\widehat{B}_{\mathrm{GCC}}(t) \leftarrow \Delta d(t), \tag{1}$$

onde $\widehat{B}_{\mathrm{GCC}}(t)$ denota a largura de banda estimada no instante t e $\Delta d(t)$ representa a variação temporal do atraso de chegada dos pacotes. Essa relação não corresponde a uma função analítica explícita; ela representa um controlador heurístico interno ao GCC, que combina detecção de tendência de atraso, estados de sobreuso/subuso e atualizações aditivas e multiplicativas de taxa para inferir a capacidade viável do enlace.

A taxa-alvo utilizada pelo sistema pode ser expressa como uma versão limitada da estimativa, incorporando uma margem de segurança e limites operacionais:

$$B_{\mathrm{alvo}}(t) = \mathrm{clip}\big(\alpha \cdot \widehat{B}_{\mathrm{GCC}}(t), B_{\min}, B_{\max}\big), \tag{2}$$

onde $B_{\mathrm{alvo}}(t)$ é a taxa aplicada no instante t, $\alpha \in (0, 1]$ define uma margem de segurança, $B_{\min}$ e $B_{\max}$ são os limites mínimo e máximo de taxa considerados, e $\mathrm{clip}(\cdot)$ representa a saturação nesses limites.

4.2. Estimador de Largura de Banda Baseado em TCP_INFO

O estimador baseado em *TCP_INFO* utiliza métricas internas do *kernel* Linux obtidas a partir de uma conexão TCP dedicada exclusivamente à sondagem da rede. Essa conexão não transporta o fluxo de vídeo, que é enviado separadamente via UDP, e tem como objetivo fornecer informações sobre o estado do congestionamento do enlace. Entre as métricas coletadas destacam-se o tamanho da janela de congestionamento, o tamanho máximo de segmento e o tempo de ida e volta.

A estimativa bruta de largura de banda é calculada a partir de uma aproximação do throughput TCP:

$$\widehat{B}_{\mathrm{bruto}}(t) = \frac{\mathrm{cwnd}(t) \cdot \mathrm{MSS} \cdot 8}{RTT(t)}, \tag{3}$$

onde $\widehat{B}_{\mathrm{bruto}}(t)$ é a estimativa instantânea de largura de banda em bits por segundo, $\mathrm{cwnd}(t)$ representa o tamanho da janela de congestionamento em número de segmentos no instante t, MSS é o tamanho máximo de segmento em bytes, o fator 8 converte bytes para bits, e $RTT(t)$ corresponde ao tempo de ida e volta medido no instante t.

Para reduzir oscilações decorrentes de ruído de medição, essa estimativa é suavizada por meio de uma média móvel exponencial:

$$\widehat{B}_{\mathrm{suave}}(t) = \beta \cdot \widehat{B}_{\mathrm{suave}}(t-1) + (1-\beta) \cdot \widehat{B}_{\mathrm{bruto}}(t), \tag{4}$$

onde $\widehat{B}_{\mathrm{suave}}(t)$ é a estimativa suavizada de largura de banda no instante t e $\beta \in (0, 1)$ é o fator de suavização que controla o peso atribuído às estimativas passadas.

A largura de banda disponível ao fluxo de vídeo é então obtida descontando-se a taxa atualmente utilizada pelo tráfego de aplicação:

$$\widehat{B}_{\mathrm{disp}}(t) = \max\left\{0,\ \widehat{B}_{\mathrm{suave}}(t) - B_{\mathrm{fg}}(t)\right\}, \tag{5}$$

onde $B_{\mathrm{fg}}(t)$ denota a taxa do tráfego de primeiro plano (vídeo).

4.3. Estimador de Largura de Banda Baseado em Goodput

O estimador baseado em *goodput* utiliza medições realizadas no receptor para inferir a capacidade efetiva percebida pela aplicação. Nesse método, o receptor mede o volume de dados corretamente recebidos ao longo de janelas temporais fixas e calcula a taxa efetiva de entrega, desconsiderando perdas e retransmissões.

A estimativa de *goodput* é calculada como:

$$\widehat{G}(t) = \frac{8 \cdot \Delta\mathrm{bytes}(t)}{\Delta t}, \tag{6}$$

onde $\widehat{G}(t)$ é a taxa efetiva de recepção em bits por segundo, $\Delta\mathrm{bytes}(t)$ representa o número de bytes recebidos corretamente durante o intervalo Δt, e o fator 8 converte bytes para bits.

O valor de $\widehat{G}(t)$ é reportado ao emissor e utilizado como uma aproximação direta da largura de banda disponível:

$$\widehat{B}_{\mathrm{disp}}(t) \approx \widehat{G}(t). \tag{7}$$

A taxa de codificação do próximo *chunk* é então selecionada a partir do *bitrate ladder*, escolhendo-se o maior nível que não exceda a estimativa reportada.

4.4. Largura de Banda Fixa

Como linha de base experimental, foi considerado um cenário sem estimação de largura de banda, no qual o emissor opera com uma taxa de codificação constante durante toda a transmissão. Nesse caso, não há realimentação entre receptor e emissor, e a largura de banda disponível não é estimada explicitamente:

$$B_{\mathrm{alvo}}(t) = B_{\mathrm{fixo}}, \quad \forall t, \tag{8}$$

onde B_{fixo} representa a taxa de codificação previamente definida. Essa linha de base corresponde, portanto, ao caso sem *Bandwidth Estimation* (BWE), no qual a taxa aplicada ao vídeo permanece fixa ao longo de todo o experimento, servindo como referência para avaliar os ganhos dos métodos adaptativos em termos de estabilidade, eficiência e qualidade percebida.

5. Metodologia e Protocolo Experimental

Os experimentos foram conduzidos em um ambiente controlado com um servidor responsável pela codificação e transmissão e um cliente responsável pela recepção e análise do fluxo. O sistema foi executado em um notebook com processador *Intel Core* i7-13650HX, GPU NVIDIA RTX 4060 e Linux 24.04. A transmissão foi realizada com *GStreamer* 1.26.0, utilizando o plugin *webrtcsink*. A visualização ocorreu em um dispositivo cliente dedicado. Para fins de reprodutibilidade, os scripts e arquivos auxiliares utilizados neste trabalho estão disponíveis em repositório público: https://github.com/publioelon/avaliacao-estimacao-de-largura-de-banda-streaming-tempo-real.

Os experimentos utilizaram o vídeo 360° *London on Tower Bridge*, adotado por Caruso *et al.* [Caruso et al. 2024], processado na resolução de 1920×960 pixels a 30

quadros por segundo. Os limites de largura de banda foram definidos a partir de um tamanho máximo por quadro $S_{\max}$, de modo que a taxa necessária para transmitir continuamente quadros desse tamanho a 30 fps seja dada por $B_{\max} = S_{\max} \cdot f_{\text{fps}}$. Assim, para $S_{\max} \approx 300\,\text{KB}$, obtém-se $72\,\text{Mb/s}$, e para $S_{\max} \approx 600\,\text{KB}$, obtém-se $144\,\text{Mb/s}$. Esses valores foram, portanto, adotados como regimes de baixa e alta largura de banda, respectivamente, por corresponderem à capacidade necessária para sustentar a transmissão de quadros de até 300 KB e 600 KB no cenário experimental considerado. O vídeo está disponível em[1].

A topologia experimental empregou um comutador virtual em que o controle de tráfego foi implementado via *traffic control* (*tc*) do Linux, ferramenta usada para configurar mecanismos de controle e limitação no enlace. Nesse contexto, utilizou-se *ingress policing*, mecanismo do *tc* para limitar a taxa de entrada de pacotes, impondo os tetos de 72 e 144 Mb/s no enlace experimental. Assim, reproduzem-se, em ambiente controlado, cenários com restrição explícita de capacidade, nos quais a banda disponível ao fluxo de vídeo é determinada pelo limite configurado e pela competição com o tráfego de fundo. Consideraram-se, portanto, dois regimes: baixa largura de banda (72 Mb/s) e alta largura de banda (144 Mb/s).

O fluxo de vídeo foi segmentado em *chunks* temporais compostos por 112 quadros, correspondendo a aproximadamente 3,7 segundos por *chunk*, considerando uma taxa de 30 quadros por segundo. Cada experimento foi composto por oito *chunks*, totalizando aproximadamente 30 segundos de transmissão contínua. Ao final de cada *chunk*, o sistema reavaliava a largura de banda disponível no enlace e ajustava a taxa de codificação do *chunk* subsequente com base nessa estimativa. Durante toda a duração do experimento, o tráfego de vídeo (*foreground*) e o tráfego de fundo (*background*) permaneceram ativos de forma contínua, competindo simultaneamente pelos recursos do enlace.

A variação de banda foi induzida pela redução progressiva do tráfego de fundo a cada transição entre *chunks*, mantendo-se fixa a capacidade do link via *ingress policing*; assim, a banda disponível ao vídeo aumentou em oito patamares, sem imposição explícita ao codificador. O vídeo foi então codificado em H.264 com o *preset* padrão do NVENC, e a taxa de cada *chunk* foi definida pela estimativa do método de controle, variando de 8,7 a 68,8 Mb/s no cenário de baixa banda (teto de 72 Mb/s) e de 69,9 a 138,3 Mb/s no cenário de alta banda (teto de 144 Mb/s), permitindo avaliar a adaptação dos estimadores às condições efetivas da rede.

Para viabilizar a codificação guiada por saliência, desenvolveu-se a ferramenta AppEncCudaROI baseada no NVIDIA Video Codec SDK[2], que utiliza o NVENC para codificação H.264 e aplica variações de QP por região a partir de mapas de saliência. O quadro é particionado em *tiles* com prioridades hierárquicas, com pesos 0,7 para regiões de maior saliência, 0,2 para regiões adjacentes e 0,1 para regiões periféricas. Esses pesos orientam o ajuste de quantização por *tile* e a redistribuição espacial do bitrate. A taxa global permanece constante e é definida pelo mecanismo de controle de taxa, independentemente do modelo de saliência adotado.

[1] `https://www.mettle.com/360vr-master-series-free-360-downloads-page/`, item *London on Tower Bridge*.

[2] `https://developer.nvidia.com/video-codec-sdk`.

Para avaliar a qualidade visual do vídeo, utilizamos as métricas *Peak Signal-to-Noise Ratio* (PSNR), *Structural Similarity Index Measure* (SSIM) e *Video Multi-Method Assessment Fusion* (VMAF). O PSNR e o SSIM foram adotados por constituírem métricas objetivas clássicas, amplamente consolidadas na literatura de compressão e avaliação de vídeo, enquanto o VMAF foi utilizado por complementar essa análise com uma medida perceptual agregada, mais alinhada à qualidade visual percebida pelo usuário. O PSNR é obtido a partir do erro quadrático médio entre o quadro de referência I e o quadro reconstruído $\hat{I}$, dado por

$$\mathrm{MSE} = \frac{1}{HW} \sum_{x,y} \big(I(x,y) - \hat{I}(x,y)\big)^2, \tag{9}$$

e

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}} \right), \tag{10}$$

onde MAX é o valor máximo possível do pixel. O SSIM quantifica a similaridade estrutural entre I e $\hat{I}$, sendo definido por

$$\mathrm{SSIM}(I, \hat{I}) = \frac{(2\mu_I \mu_{\hat{I}} + C_1)(2\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)}, \tag{11}$$

em que μ e σ representam estatísticas locais (média e desvio padrão), $\sigma_{I\hat{I}}$ é a covariância e C_1, C_2 são constantes de estabilização. O VMAF é empregado como um índice perceptual agregado que combina múltiplos descritores de qualidade por meio de fusão aprendida, sendo reportado diretamente como escore final.

Como o objetivo é comparar *codificação guiada por saliência* e *codificação uniforme* em termos do impacto nas regiões perceptualmente relevantes, além das métricas globais reportamos versões ponderadas de PSNR e SSIM em nível de *tile*. Seja $\mathcal{T}$ o conjunto de *tiles*, com pesos w_t atribuídos conforme a relevância perceptual (por exemplo, $w_t \in \{0.7, 0.2, 0.1\}$). Definimos o erro ponderado como

$$\mathrm{MSE}_w = \frac{\sum_{t \in \mathcal{T}} w_t \, \mathrm{MSE}_t}{\sum_{t \in \mathcal{T}} w_t}, \tag{12}$$

e o PSNR ponderado como

$$\mathrm{PSNR}_w = 10 \log_{10} \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}_w} \right). \tag{13}$$

De forma análoga, definimos o SSIM ponderado por *tile* como

$$\mathrm{SSIM}_w = \frac{\sum_{t \in \mathcal{T}} w_t \, \mathrm{SSIM}_t}{\sum_{t \in \mathcal{T}} w_t}. \tag{14}$$

Dessa forma, regiões com maior probabilidade de atenção contribuem proporcionalmente mais para o escore, permitindo uma comparação direta entre *com saliência* e *sem saliência* sem alterar a taxa global imposta pelo mecanismo de controle de taxa.

Sobre essa infraestrutura, foram avaliados métodos de estimação de largura de banda. No Google Congestion Control, o servidor atuou como emissor e o cliente como

receptor, enquanto um tráfego de fundo competiu pela capacidade do enlace. O controle baseou-se nos relatórios TWCC enviados periodicamente pelo receptor, que descrevem os tempos de chegada dos pacotes. Com esses relatórios, o emissor estimou a banda disponível e ajustou dinamicamente a taxa de codificação do vídeo.

No método baseado em TCP_INFO, o servidor manteve uma conexão TCP dedicada exclusivamente à sondagem da rede, independente do fluxo de vídeo. Essa conexão permitiu coletar métricas como taxa de entrega, tempo de ida e volta e estado da janela de congestionamento, as quais foram utilizadas para estimar a largura de banda disponível. O tráfego de fundo permaneceu ativo durante todo o experimento, garantindo competição realista por recursos e permitindo avaliar a capacidade do estimador em acompanhar variações de carga.

Por fim, no método baseado em *goodput*, o cliente foi responsável por medir a taxa efetiva de dados recebidos ao longo de janelas temporais fixas. Essa taxa foi então reportada ao emissor, que ajustou o bitrate do próximo chunk de acordo com a capacidade observada. Por refletir diretamente o desempenho percebido pela aplicação, esse método apresentou elevada sensibilidade a variações rápidas da rede, permitindo avaliar a eficácia do controle adaptativo sob condições dinâmicas.

6. Resultados e Discussão

Os resultados são avaliados por PSNR, SSIM, VMAF e tamanho do arquivo por *chunk*, em função da taxa de bits aplicada em baixa e alta largura de banda, definida pela estimativa de cada método. Assim, cada ponto nos gráficos representa um *chunk*: a taxa selecionada e seu impacto na qualidade e no tamanho do arquivo.

Na Figura 2 (cenário de baixa largura de banda), os estimadores de largura de banda como o GCC+TWCC e *goodput*, exceto o caso de *overshoot* do TCP_INFO, atingem resultados próximos aos do modo Fixo, sem *Bandwidth Estimation* (BWE), com PSNR entre 54 e 56, SSIM entre 0,997 e 1,0 e VMAF entre 97,5 e 98,5. Ainda que, em alguns pontos, isso ocorra ao custo de um aumento incipiente no tamanho do arquivo, observa-se que o emprego de estimadores de largura de banda preservou a qualidade dos quadros em momentos de largura de banda limitada.

Em adição, os estimadores de largura de banda exibem desempenho próximo conforme a banda estimada disponível é menor no modo baixa largura de banda até 30 Mbps, e passam a se diferenciar conforme a banda aumenta. O que evidencia que os estimadores de largura de banda são mais eficientes na preservação da qualidade da imagem durante a compressão quando há valores reduzidos de banda disponível.

Logo, no cenário de baixa largura de banda e com valor fixo (sem BWE), a codificação guiada por saliência mantém as métricas resultantes com valores próximos. No modo com valor fixo de largura de banda disponível, o PSNR médio foi de 53,61 dB sem saliência e 54,68 dB com saliência; o SSIM foi de 0,9975 e 0,9979; e o VMAF foi de 97,94 e 97,98. Em contrapartida, observou-se aumento do tamanho do *chunk*, com incremento de 43,6% no tamanho do arquivo.

Na Figura 3 (cenário de alta largura de banda), observa-se que o tamanho de arquivo produzido em decorrência de cada estimador de largura de banda permaneceu abaixo dos valores do cenário em que não há o uso de estimadores (Fixo/Sem BWE).

Assim, à medida que a banda disponível aumenta, o uso de estimadores mantém baixo o tamanho do *chunk* e, simultaneamente, preservam a qualidade de imagem nas métricas de PSNR, SSIM e VMAF, evidenciando um uso mais eficiente do bitrate. Esse comportamento é desejável em um sistema de *streaming* de vídeo 360° em tempo real, pois reduz o volume transmitido sem degradar a qualidade visual e tende a melhorar a estabilidade sob competição por capacidade no enlace.

Não obstante, as métricas entram em regime de saturação e o *goodput* sem saliência mantém os maiores valores de PSNR e SSIM, com VMAF em torno de 98,0–98,3 e incremento moderado no tamanho do arquivo. Nesse cenário, o GCC+TWCC sem saliência tende a superar a taxa fixa sem saliência em aproximadamente 0,6–1,0 dB de PSNR e 0,3–0,5 pontos de VMAF, com menor tamanho de arquivo, refletindo maior reatividade às variações de capacidade.

Conforme as Figuras 2 e 3, o estimador *goodput* se distingue porque a largura de banda estimada em um determinado *chunk* é definida a partir da banda estimada no *chunk* anterior e retorna ao servidor como *feedback*. Assim, o servidor ajusta a estimativa do *chunk* seguinte com base nesse valor, introduzindo atraso na medição seguinte. Em vista disso, a largura de banda estimada tende a acompanhar o valor anterior: em baixa largura de banda, a curva progride aproximadamente de 9,0 Mb/s a 68,8 Mb/s, enquanto, em alta largura de banda, progride de 69,9 Mb/s a 138,3 Mb/s.

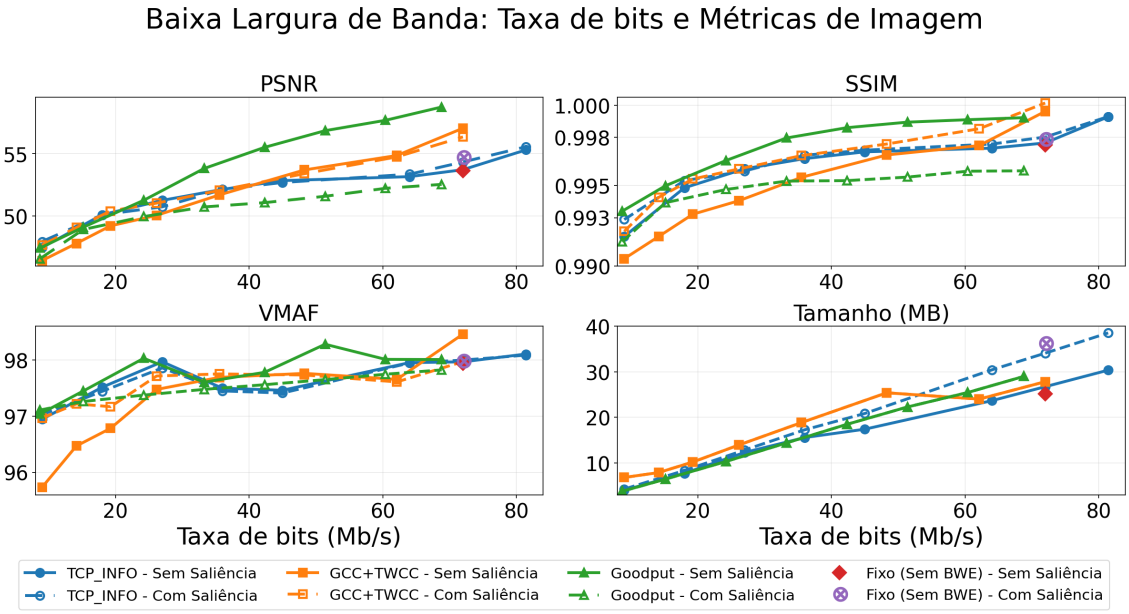

Figura 2. Baixa largura de banda: métricas de imagem em função da taxa de bits selecionada a partir da largura de banda estimada por cada método.

Em adição, os métodos propostos apresentaram comportamentos de estimação distintos ao longo dos experimentos, com o GCC+TWCC operando com maior margem em relação ao limite de banda imposto, o *goodput* ajustando-se de forma gradual a partir da taxa efetivamente entregue ao receptor e o TCP_INFO exibindo tendência a selecionar patamares mais elevados, refletida no aumento do tamanho do arquivo em parte do regime avaliado. Além disso, os métodos GCC+TWCC com e sem saliência, assim como Goodput com e sem saliência, produziram *chunks* com praticamente o mesmo tamanho de

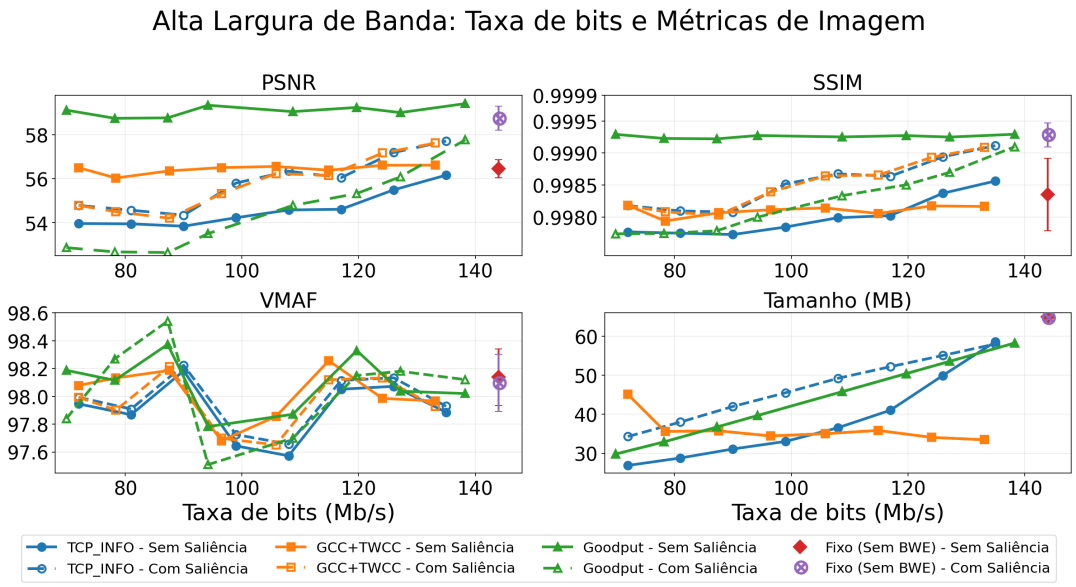

Figura 3. Alta largura de banda: métricas de imagem em função da taxa de bits selecionada a partir da largura de banda estimada por cada método.

arquivo, de modo que as curvas de tamanho ficam sobrepostas.

Na Figura 4, o GCC+TWCC subestima a capacidade nos *chunks* iniciais, com erros de -2,9 a -5,5 Mb/s entre os chunks 1 e 4, permanecendo negativo no chunk 5 com erro de -0,7 Mb/s, e passa a superestimar nos chunks 6 e 7 com erros de 5,1 e 7,0 Mb/s, caracterizando inversão de viés e maior sensibilidade às transições de carga. O método baseado em goodput ajusta-se gradualmente e mantém erro de menor magnitude, variando de -1,8 a 3,8 Mb/s ao longo do experimento, o que indica melhor aderência à capacidade efetiva sob restrição de banda. Já o TCP_INFO apresenta crescimento progressivo do erro até o chunk 4, com valores de 1,0 a 4,0 Mb/s entre os chunks 1 e 4, e passa a superestimar de forma acentuada a partir do chunk 5, atingindo 15,0 Mb/s nos chunks 5 e 6 e 16,4 Mb/s no último chunk, evidenciando viés persistente de superestimação e maior risco de *overshoot* em baixa banda.

Na Figura 5, o GCC+TWCC inicia com subestimação, com erros negativos de -1,6 e -0,5 Mb/s nos chunks 1 e 2, passa a superestimar a partir do chunk 3 com erro de 0,5 Mb/s e mantém superestimação crescente até o fim do experimento, com erros de 1,8, 2,9, 4,1 e 5,2 Mb/s nos chunks 4 a 7, mantendo variações controladas no regime de maior capacidade. O método baseado em goodput apresenta oscilações mais marcantes: começa com erros negativos de -2,1, -1,8, -0,7 e -1,8 Mb/s nos chunks 0 a 3 e, após a transição, passa a superestimar com erros de 4,7, 7,7, 7,2 e 10,3 Mb/s nos chunks 4 a 7, sugerindo maior sensibilidade ao atraso de realimentação e à variabilidade após mudanças de capacidade. Já o TCP_INFO mantém crescimento aproximadamente linear do erro ao longo dos chunks, com valores de 1,0 a 7,0 Mb/s entre os chunks 1 e 7, indicando tendência sistemática a superestimar mesmo em alta banda.

A Tabela 1 quantifica as diferenças médias de PSNR, SSIM, VMAF e tamanho do arquivo entre a codificação com saliência e a codificação uniforme, enquanto a Tabela 2 sintetiza propriedades dinâmicas dos estimadores que auxiliam a interpretar esses resul-

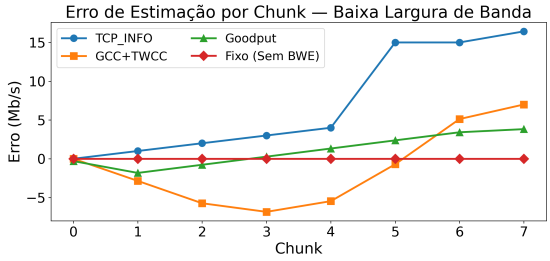

Figura 4. Erro de estimação de largura de banda por *chunk* no cenário de baixa largura de banda.

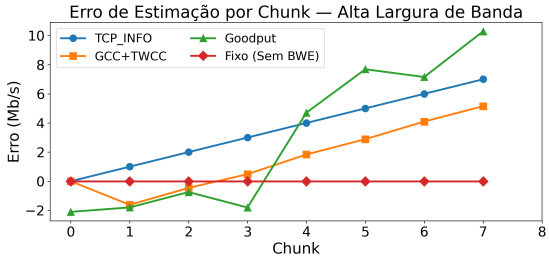

Figura 5. Erro de estimação de largura de banda por *chunk* no cenário de alta largura de banda.

Tabela 1. Diferença média de PSNR, SSIM, VMAF e tamanho do arquivo entre a codificação com e sem saliência, nos cenários de baixa e alta largura de banda, por método de estimação.

Método	**Cenário**	ΔPSNR (dB)	ΔSSIM	ΔVMAF	ΔTam. (%)
TCP_INFO	Baixa	0.11	0.000 29	−0.02	21.0
TCP_INFO	Alta	1.25	0.000 53	0.06	22.3
GCC+TWCC	Baixa	0.49	0.001 49	0.26	0.0
GCC+TWCC	Alta	−0.70	0.000 40	−0.06	0.0
Goodput	Baixa	−3.34	−0.002 59	−0.27	0.0
Goodput	Alta	−4.64	−0.001 02	0.00	0.0
Fixo (Sem BWE)	Baixa	1.07	0.000 37	0.04	43.6
Fixo (Sem BWE)	Alta	2.30	0.000 93	−0.04	−0.3

tados em termos de estabilidade, reatividade e sensibilidade a ruído. Na Tabela 1, as variações de PSNR, SSIM e VMAF correspondem à diferença entre as médias por *chunk* com saliência e sem saliência, promediadas ao longo de oito *chunks*, e a variação do tamanho do arquivo é essa mesma diferença expressa de forma relativa, normalizada pelo valor uniforme e apresentada em porcentagem.

Adicionalmente, a codificação guiada por saliência tende a elevar PSNR e SSIM quando combinada com TCP_INFO, tanto em baixa quanto em alta largura de banda, ao custo de aumento no tamanho do arquivo. Em baixa largura de banda, observa-se ganho também com GCC+TWCC, sem aumento de tamanho, enquanto no modo fixo os ganhos de qualidade vêm acompanhados de variação no tamanho do *chunk*. Em contraste, para o método baseado em *goodput* e para o GCC+TWCC em alta largura de banda, as diferenças de qualidade são pequenas ou negativas, mantendo o mesmo tamanho de arquivo, o que sugere margem para otimizar a alocação espacial de bits nesse regime.

A Tabela 2 apresenta atributos dinâmicos associados aos estimadores. O GCC+TWCC exibe maior reatividade e menor erro médio absoluto global, o TCP_INFO fornece um sinal mais estável, e o método baseado em *goodput* é mais sensível a ruído e ao atraso de medição. Essas características se refletem nas diferenças de erro de estimação e nas métricas de qualidade, reforçando que a estimação define o taxa temporal de bits por *chunk*, enquanto a codificação guiada por saliência redistribui essa taxa espacialmente.

Tabela 2. Características qualitativas dos métodos de estimação de largura de banda.

Método	Reatividade	Estabilidade	Sensibilidade a ruído	Adequação a RTC
GCC+TWCC	Alta	Média	Média	Alta
TCP_INFO	Média	Alta	Baixa	Média
Goodput	Baixa	Média	Alta	Baixa

Os resultados indicam que a estimação de largura de banda ajusta a taxa por *chunk* à capacidade efetiva do enlace, preservando a qualidade em baixa largura de banda e, em alta largura de banda, contendo o tamanho do arquivo sem degradar PSNR, SSIM e VMAF, o que evidencia maior eficiência no uso de bits. O GCC+TWCC foi o mais reativo às variações de capacidade, o *goodput* ajustou-se de forma gradual com atraso de realimentação, e o TCP_INFO apresentou tendência a superestimação e *overshoot*, sobretudo em baixa largura de banda. A codificação guiada por saliência elevou PSNR e SSIM em parte dos cenários, com ganhos dependentes do estimador e, ocasionalmente, aumento do tamanho do *chunk*. Como limitação, os experimentos utilizaram um único *preset* e pesos fixos de saliência, sem otimização por configuração.

7. Conclusão

O presente trabalho investigou o controle adaptativo de taxa em *streaming* de vídeo 360° em tempo real a partir de estimadores de largura de banda, considerando que arquiteturas convencionais não fornecem ao codificador um sinal explícito da capacidade disponível. Avaliou-se a compressão com taxa por *chunk* definida por diferentes estimadores e comparou-se essa adaptação temporal com uma codificação por *tiles* guiada por saliência, que redistribui espacialmente o bitrate sem aumentar a taxa global. Como contribuição secundária, propõem-se dois estimadores externos ao *pipeline* de compressão—TCP_INFO e *goodput*—além dos baselines GCC+TWCC e taxa fixa. Os resultados indicam que a estimação melhora o ajuste entre taxa aplicada e capacidade utilizável, reduzindo sobrecarga e subutilização, e que a saliência pode priorizar regiões perceptualmente relevantes com impacto controlado no tamanho do *chunk*; em particular, os estimadores preservam a qualidade quando a banda é limitada e, quando há maior disponibilidade, contêm o crescimento do *chunk* mantendo qualidade equivalente. Como limitação, utilizou-se um único *preset* e pesos fixos de saliência; como trabalhos futuros, pretende-se empregar aprendizagem por reforço para otimizar parâmetros de compressão e reduzir o compromisso entre qualidade e tamanho do arquivo.

Agradecimentos

Este trabalho foi apoiado pela Ericsson Telecomunicações Ltda. e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), por meio do projeto 2021/00199-8, CPE SMARTNESS.

References

Carlucci, G., De Cicco, L., Holmer, S., and Mascolo, S. (2016). Analysis and design of the google congestion control for web real-time communication. In *Proceedings of*

the 7th International Conference on Multimedia Systems (MMSys '16), pages 1–12, Klagenfurt, Austria. ACM.

Caruso, A., Grasso, C., Raftopoulos, R., and Schembra, G. (2024). An Adaptive Closed-Loop Encoding VNF for Virtual Reality Applications. In *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*, pages 80–88. IEEE.

Fouladi, S., Emmons, J., Orbay, E., Wu, C., Wahby, R. S., and Winstein, K. (2018). Salsify: Low-latency network video through tighter integration between a video codec and a transport protocol. In *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 267–282. USENIX Association.

Jiang, W., Han, B., Habibi, M. A., and Schotten, H. D. (2021). The road towards 6g: A comprehensive survey. *IEEE Open Journal of the Communications Society*, 2:334 to 366.

Jiang, X., Naas, S. A., Chiang, Y.-H., Sigg, S., and Ji, Y. (2020). SVP: Sinusoidal Viewport Prediction for 360-Degree Video Streaming. *IEEE Access*, 8:164471–164480.

Khan, M. A., Baccour, E., Chkirbene, Z., Erbad, A., Hamila, R., Hamdi, M., and Gabbouj, M. (2022). A survey on mobile edge computing for video streaming: Opportunities and challenges. *IEEE Access*, 10:120514–120547.

Li, M., Wu, Y.-L., and Chang, C.-R. (2014). Available bandwidth estimation for network paths with multiple tight links and bursty traffic. *Computer Networks*, 72:16–30.

Nguyen, D. V., Tran, H. T. T., Pham, A. T., and Thang, T. C. (2019). An Optimal Tile-Based Approach for Viewport-Adaptive 360-Degree Video Streaming. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):29–42.

Nguyen, D. V., Tran, H. T. T., and Thang, T. C. (2020). An Evaluation of Tile Selection Methods for Viewport-Adaptive Streaming of 360-Degree Video. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(1).

Wang, S., Yang, S., Li, H., Zhang, X., Zhou, C., Xu, C., Qian, F., Wang, N., and Xu, Z. (2022). Salientvr: Saliency-driven mobile 360-degree video streaming with gaze information. In *Proceedings of the 28th Annual International Conference on Mobile Computing and Networking (MobiCom '22)*, Sydney, NSW, Australia. ACM.

Xie, L., Xu, Z., Ban, Y., Zhang, X., and Guo, Z. (2017). 360ProbDASH: Improving QoE of 360 Video Streaming Using Tile-based HTTP Adaptive Streaming. In *Proceedings of the 25th ACM International Conference on Multimedia (MM '17)*, pages 315–323.

Zhang, J., Wang, J., Jiang, H., and Zhang, Z.-L. (2019). Learning to coordinate video codec and transport protocol for mobile video telephony. In *Proceedings of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–15. ACM.

Zhang, Y., Chen, Z., Wang, Y., and Liu, F. (2023). Bridging the gap between qoe and qos in congestion control. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. USENIX Association.