



# Avaliação de políticas de adaptação de taxa de bits para sessões de vídeo 360°

*F. Ricci   G. Mariani   D. Velasco   N. Fonseca   C. Melo*

Technical Report - IC-22-23 - Relatório Técnico  
June - 2022 - Junho

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

# Avaliação de políticas de adaptação de taxa de bits para sessões de vídeo 360°

Fabio Ricci\*   Gabriel Mariani†   Daniela Velasco‡   Nelson Fonseca§   Cesar Melo¶

30/04/2022

## Resumo

Este trabalho discorrerá sobre avaliação de políticas de adaptação de taxa de transmissão para sessões de vídeo 360°. A proposta deste é o entendimento dos principais algoritmos de adaptação de taxa de transmissão para vídeos regulares e como pode-se explorar a algoritmos atuais para vídeos 360°.

As tecnologias que foram exploradas neste trabalho são o protocolo de rede QUIC que permite conexões multi-stream, vídeos 360°, e que tem se tornando popular ao longo dos últimos anos e algoritmos de adaptação de taxa de transmissão bem difundidos.

Foi feita uma análise qualitativa de como essas tecnologias de vídeos tradicionais se comportam em cenários de vídeos 360°. Pode-se concluir que com algumas adaptações, é possível a utilização dos algoritmos apresentados nesse novo paradigma.

## 1 Introdução

Desde sua introdução, o protocolo QUIC têm se tornado cada vez mais adotado, presente em diversos *browsers* e sendo o *default* em varias aplicações hoje em dia.

Dentre essas aplicações, uma das de maior potencial para se beneficiar das vantagens desse protocolo é o *streaming* e reprodução de vídeos, por serem muito sensíveis a largura de banda e latência, tornando o QUIC um forte candidato a se tornar o protocolo padrão nesse meio.

A transmissão de vídeos têm sido alvo de estudo, sendo um tema estudado pela academia desde os primórdios da Internet. Com o tempo, surgiram diversas técnicas de adaptação da qualidade de transmissão de vídeo (*bitrate*), baseadas em parâmetros como qualidade de canal, ocupação de Buffer, largura de banda disponível, entre outros. Hoje, essas técnicas são capazes de otimizar a transmissão de vídeo de diversas maneiras, melhorando a qualidade de experiencia.

Nesse contexto, enquadram-se vídeos 360°, os quais permitiram a visualização do conteúdo em todas as direções de uma gravação, principalmente em ambientes de realidade virtual (VR) e realidade aumentada (AR). São muito similares aos vídeos convencionais, sendo separados em segmentos (fatias temporais do vídeo) e a grande diferença é que cada segmento de vídeo 360° é

---

\*Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. [f170781@dac.unicamp.br](mailto:f170781@dac.unicamp.br).

†Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. [g197470@dac.unicamp.br](mailto:g197470@dac.unicamp.br).

‡Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. [dacasasv@gmail.com](mailto:dacasasv@gmail.com).

§Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. [nfonseca@ic.unicamp.br](mailto:nfonseca@ic.unicamp.br).

¶Inst. de Computação, UFAM, 69080-900 Manaus, AM. [cavmelo@icomp.ufam.edu.br](mailto:cavmelo@icomp.ufam.edu.br).

separado em frações independentes (*tiles*) que, em conjunto, completam a imagem em torno de um ponto central (foco da visão).

O ponto central deste trabalho é discorrer sobre *streaming* e reprodução de vídeos 360°, bem como o comportamento de algoritmos de adaptação de taxa de transmissão utilizados para vídeos comuns, aplicados à essa nova categoria de mídia. Finalmente, são propostas possíveis otimizações à essas técnicas para os cenários com vídeos 360°.

## 2 O protocolo QUIC

O QUIC é um protocolo de rede da camada de transporte, introduzido em 2012 pelo Google, que foi idealizado como alternativa ao protocolo TCP, utilizado na maior parte da internet e *web* da época. Atualmente está presente em mais da metade de todas as conexões dos *browsers Google Chrome* para os servidores do Google, além de também ser implementado nos *browsers Microsoft Edge, Firefox e Safari*.

Diferente do TCP, o QUIC foi construído sobre o protocolo UDP, porém conta com todos os benefícios do primeiro, como garantia de entrega de pacotes e controle de congestionamento como podemos observar na **Figura 1**. Isso é possível com a multiplexação de conexões entre dois terminais utilizando UDP (fazendo grande uso da multiplexação de conexões do protocolo HTTP/2), possibilitando múltiplas *streams* (**Figura 3**) de dados chegarem aos destinos finais independente umas das outras, e portanto, ignorando perda de pacotes de outras *streams*. Em contraste, o HTTP/2 rodando sobre TCP pode ter desempenho reduzido devido à atrasos *head of line blocking* se houver perda ou atraso de pacotes em alguma das *streams* multiplexadas.

Como objetivo secundário, o QUIC também reduz a latência de conexão, transporte de pacotes e estimação de uso de banda, evitando congestionamento. Além disso, ele é implementado em nível de memória de aplicação (**Figura 2**), ao invés de em nível de *kernel*, proporcionando evoluções mais rápidas de protocolos que o utilizam.

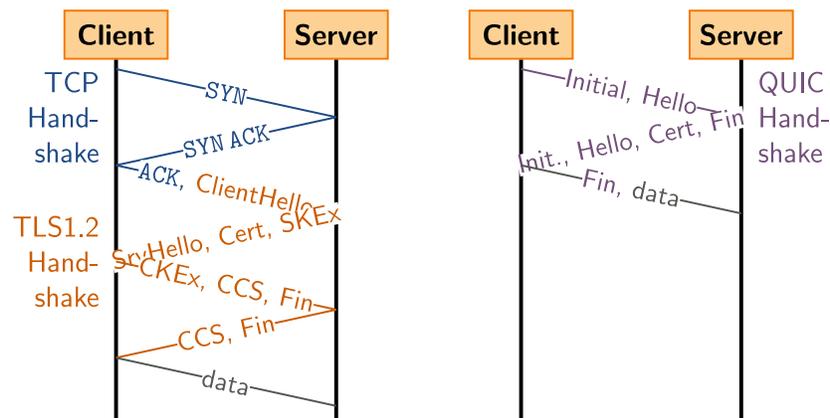


Figura 1: Comparação do handshake do QUIC vs TCP

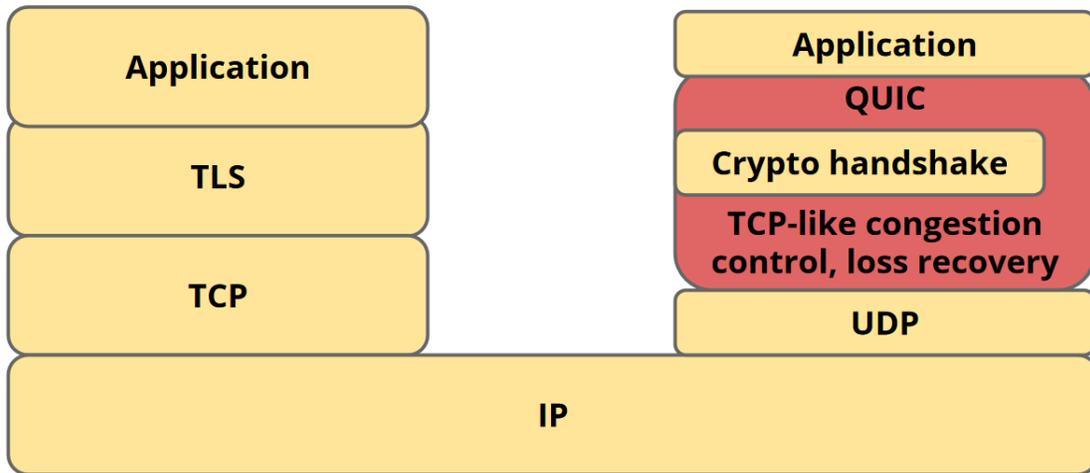


Figura 2: Comparação das camadas de aplicação e transporte QUIC vs TCP

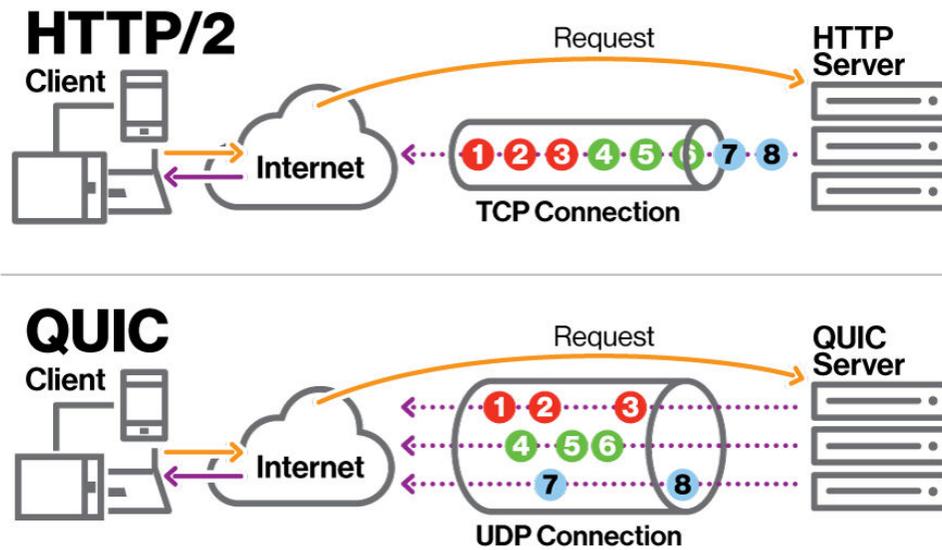


Figura 3: Comparação entre conexões com servidores QUIC vs TCP (HTTP/2)

### 3 Vídeos 360°

Vídeos podem ser divididos em pequenos trechos chamados segmentos, frações de vídeo de alguns segundos que formam o vídeo original completo. Esses segmentos são arquivos que possuem diferentes taxas de transmissão disponíveis, de forma que cada segmento pode ser requisitado em uma qualidade de acordo com as condições de rede e necessidades do cliente.

*Streaming* de vídeos 360° possui algumas diferenças em relação ao *streaming* de vídeos tradicionais. Como a área de visão (Field of View ou FOV)(**Figura 4**) de vídeos 360° é menor que a área de vídeo total disponível. Cada segmento é sub-dividido em pequenos quadros chamados *tiles*, que produzem um vídeo que compõe uma parcela do todo. Dessa forma, pode-se priorizar certos *tiles* de um segmento (apenas aqueles dentro do FOV, por exemplo) com qualidade mais alta, e deixar outras partes com qualidade mais baixa.

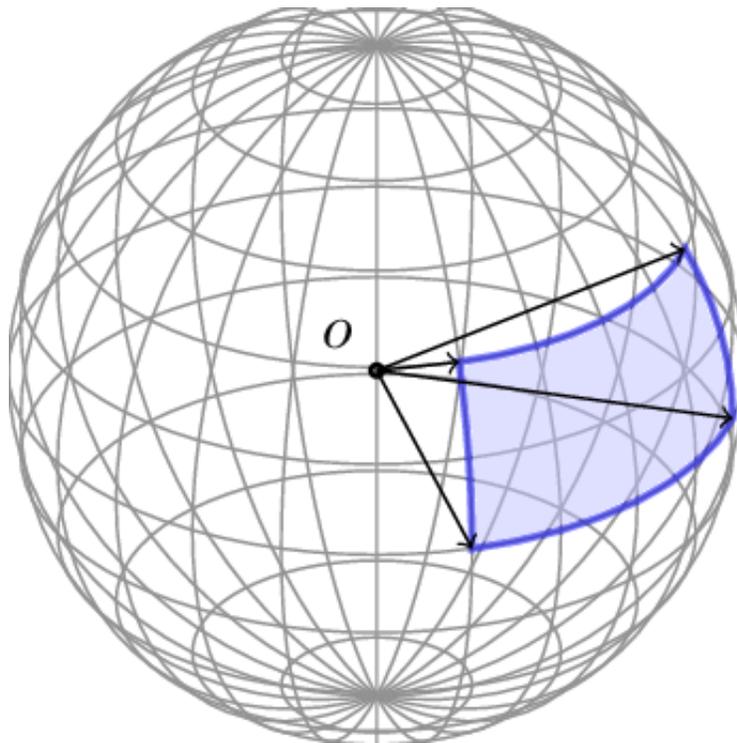


Figura 4: Visualização de um tile em um vídeo 360°

### 4 Algoritmos de Taxa de transmissão

A taxa de transmissão de um vídeo, é a principal métrica para indicar qualidade desse tipo de mídia, de forma que quanto maior a taxa, maior é o tempo de *download* e o consumo de banda.

Adaptação de taxa de transmissão é a técnica utilizada para determinar a qualidade de cada segmento de vídeo transmitido ao cliente sob certas condições de rede. O objetivo de tais algoritmos é evitar a interrupção da reprodução de mídias, mantendo-se a melhor qualidade de vídeo possível.

Os algoritmos descritos funcionam baseados em funções contínuas que levam em consideração métricas de rede (ex: vazão média de um canal, tempo de *download*, etc...). No entanto, *players* de vídeo possuem apenas algumas resoluções pré-determinadas, que muitas vezes são diferentes das taxas resultantes desses algoritmos. Dessa forma, é necessária uma técnica de mapeamento

das taxas de transmissão calculadas com as disponíveis, podendo-se aplicar uma abordagem mais conservadora (ex: arredondar para a próxima menor taxa disponível) ou mais agressiva (ex: arredondar para a próxima maior taxa). Entre essas, considera-se a abordagem conservadora mais prudente para a maioria das aplicações, já que nessa área, a experiência do usuário é prioritária, de modo que se deve evitar o travamento acima de tudo.

#### 4.1 Ocupação do Buffer

O algoritmo de Ocupação de Buffer baseia-se no **buffer** do cliente para decidir a taxa de transmissão de cada segmento. Se o *buffer* tiver pouco tempo de reprodução carregado, é necessário reduzir a qualidade para minimizar o tempo de *download* e evitar um possível travamento. Caso o **buffer** tenha muito tempo de reprodução carregado, pode-se aumentar a qualidade de cada segmento sem que haja risco de travamento.

Para determinar a taxa de transmissão, de acordo com a descrição acima, o algoritmo utiliza os seguintes parâmetros:

- *throughput*: vazão média de um canal de rede
- *buffer\_level*: Ocupação atual (em unidades de tempo, e.g. segundos) do *buffer* do cliente de segmentos baixados mas ainda não reproduzidos
- *fragment\_duration*: duração do segmento do vídeo (em unidades de tempo)
- *INSUFFICIENT\_BUFFER\_SAFETY\_FACTOR*: Parâmetro de controle da agressividade da vazão medida no canal de rede, e.g. 0.5 foi usado na implementação do algoritmo.

Por fim, a **taxa de transmissão** é definida como:

$$bitrate = throughput \times \left( \frac{buffer\_level}{fragment\_duration} \right) \times INSUFFICIENT\_BUFFER\_SAFETY\_FACTOR$$

e mapeado para a menor *taxa de transmissão* disponível mais próxima.

#### 4.2 Vazão do Canal

O algoritmo de vazão de canal baseia-se em alguns parâmetros de rede para decidir a taxa de transmissão de cada segmento. Esse algoritmo utiliza os dados dos últimos 4 segmentos baixados, fazendo uma média exponencial ponderada entre eles. Assim, é possível identificar se a qualidade está piorando ou melhorando e ajustar a taxa de transmissão de acordo.

Para determinar a taxa de transmissão, de acordo com a descrição acima, o algoritmo utiliza os seguintes parâmetros:

- $W_n$ : Tempo de *download* do segmento  $n$
- $\delta_n$ : Vazão instantânea medida pelo acesso ao segmento  $n$

Baseado nisso, calcula-se:

$$\alpha_{fast_n} = 0.5^{W_n}$$

$$\alpha_{slow_n} = 0.5^{\frac{W_n}{2}}$$

$$\mu_{fast_n} = (1 - \alpha_{fast_n}) \times \delta_{n-1} + \alpha_{fast_n} \times \mu_{fast_{n-1}}$$

$$\mu_{slow_n} = (1 - \alpha_{slow_n}) \times \delta_{n-1} + \alpha_{slow_n} \times \mu_{slow_{n-1}}$$

$$bitrate_n = \min(\mu_{fast_n}, \mu_{slow_n})$$

Dessa forma, o cálculo da qualidade do  $n$ -ésimo taxa de transmissão se dá pela tabela:

Segmento	$W$	$\delta$	$\alpha_{fast}$	$\alpha_{slow}$	$\mu_{fast}$
$n - 5$	0	0	—	—	0
$n - 4$	$W_{n-4}$	$\delta_{n-4}$	$0.5W_{n-5}$	$0.5 \frac{W_{n-5}}{2}$	0
$n - 3$	$W_{n-3}$	$\delta_{n-3}$	$0.5W_{n-4}$	$0.5 \frac{W_{n-4}}{2}$	$(1 - \alpha_{fast_{n-3}}) \times \delta_{n-4} + \alpha_{fast_{n-3}} \times \mu_{fast_{n-4}}$
$n - 2$	$W_{n-2}$	$\delta_{n-2}$	$0.5W_{n-3}$	$0.5 \frac{W_{n-3}}{2}$	$(1 - \alpha_{fast_{n-2}}) \times \delta_{n-3} + \alpha_{fast_{n-2}} \times \mu_{fast_{n-3}}$
$n - 1$	$W_{n-1}$	$\delta_{n-1}$	$0.5W_{n-2}$	$0.5 \frac{W_{n-2}}{2}$	$(1 - \alpha_{fast_{n-1}}) \times \delta_{n-2} + \alpha_{fast_{n-1}} \times \mu_{fast_{n-2}}$
$n$	—	—	$0.5W_{n-1}$	$0.5 \frac{W_{n-1}}{2}$	$(1 - \alpha_{fast_n}) \times \delta_{n-1} + \alpha_{fast_n} \times \mu_{fast_{n-1}}$

Segmento	$\mu_{slow}$
$n - 5$	0
$n - 4$	0
$n - 3$	$(1 - \alpha_{slow_{n-3}}) \times \delta_{n-4} + \alpha_{slow_{n-3}} \times \mu_{slow_{n-4}}$
$n - 2$	$(1 - \alpha_{slow_{n-2}}) \times \delta_{n-3} + \alpha_{slow_{n-2}} \times \mu_{slow_{n-3}}$
$n - 1$	$(1 - \alpha_{slow_{n-1}}) \times \delta_{n-2} + \alpha_{slow_{n-1}} \times \mu_{slow_{n-2}}$
$n$	$(1 - \alpha_{slow_n}) \times \delta_{n-1} + \alpha_{slow_n} \times \mu_{slow_{n-1}}$

E mapeado para a menor taxa de transmissão disponível mais próximo.

### 4.3 BOLA (Buffer Occupancy based Lyapunov Algorithm)

BOLA é um algoritmo de Ocupação de Buffer que faz uso dos métodos de Lyapunov. Esse algoritmo é amplamente usado na indústria pelo MPEG-DASH e também por diversos *players* de vídeo. O principal objetivo deste algoritmo é maximizar a qualidade do segmento evitando travamentos.

#### 4.3.1 Otimização de Lyapunov

A otimização de Lyapunov é o uso da função Lyapunov para controlar de forma otimizada um sistema dinâmico. As funções de Lyapunov são muito usadas na teoria de controle para garantir a estabilidade do sistema. O algoritmo BOLA faz uso destas funções para determinar a qualidade de vídeo.

#### 4.3.2 BOLA-Basic

Diferentemente do próprio BOLA, o BOLA-Basic é a implementação muito simplificada. A qualidade é dada em função do tamanho e da importância do segmento, além de dois parâmetros de controle. Caso o buffer não estiver cheio, o algoritmo escolhe o próximo segmento com mais qualidade, caso contrario não é feito *download* de nenhum segmento até a próxima tomada de decisão. Pode-se ver o comportamento descrito na **Figura 5** abaixo:

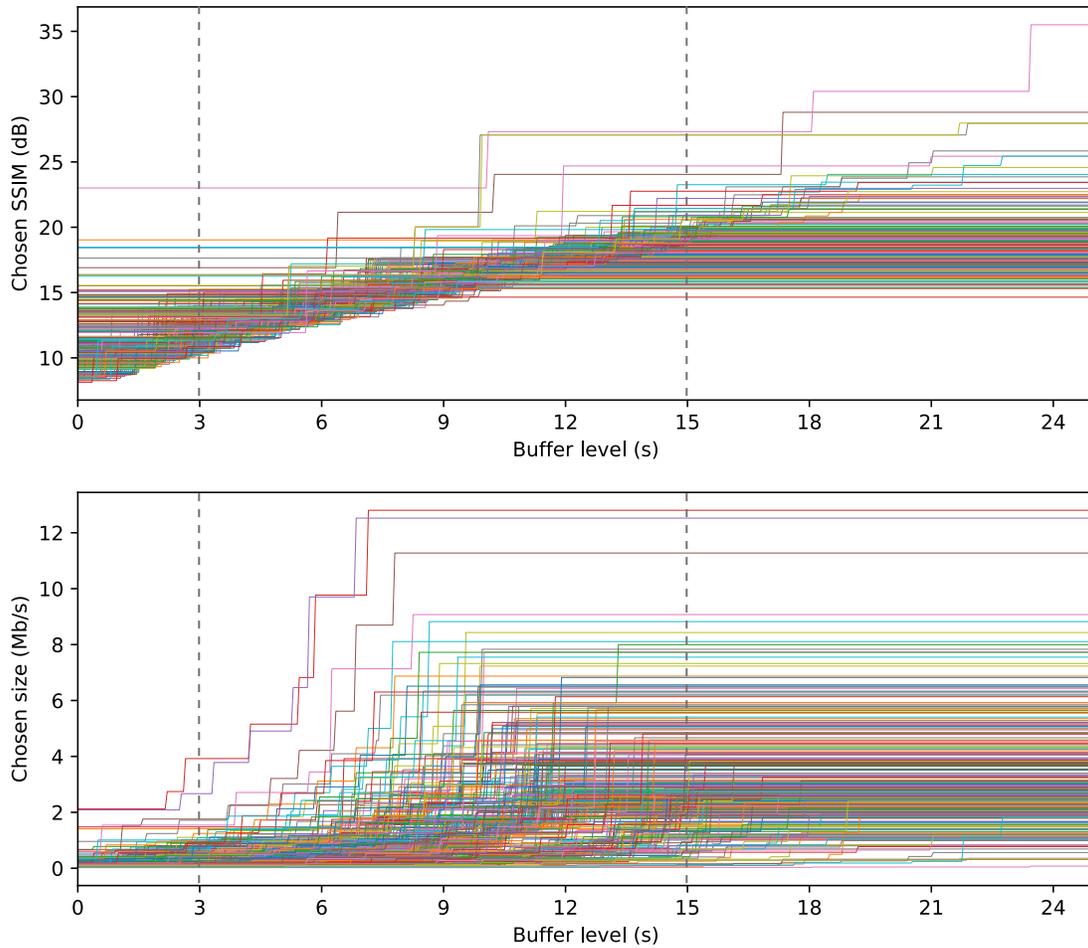


Figura 5: limiar decisão BOLA-BASIC em um vídeo de 10 minutos. Cada linha representa um único segmento de 120 frames. linha pontilhada delimita min/max buffer level.

## 5 Melhorias Propostas

Apesar dos algoritmos descritos anteriormente funcionarem para vídeos tradicionais, suas aplicações mudam para os vídeos 360°. Nessa seção, são propostas alterações nos algoritmos para uso em vídeos 360°.

A primeira melhoria proposta é a adaptação dos algoritmos para esses novos tipos de mídias, com as focando na prioridade do campo de visão (FOV) e na capacidade de execução no cenário *multi-tile* dos vídeos 360°. O primeiro passo para isso, é permitir que essas estratégias gerenciem mais de um segmento ao mesmo tempo, ou seja, possa ser feito o *download* de diferentes *frames* intercaladamente, priorizando-se o FOV atual do usuário. Para isso, são necessárias poucas mudanças na lógica dos algoritmos, porém agora as estruturas de dados precisam levar em consideração que o vídeo é união de vários segmentos simultâneos, porém direcionalmente distintos.

Igualmente importante, é priorização dos *tiles* no campo de visão, já que capacidade de usar esses dados na decisão de qualidade é extramente importante. Os *tiles* que estão fora do campo de visão, não serão vistos no segmento atual (e muito provavelmente nos próximos) que está sendo reproduzido, de forma que a importância e prioridade de qualidade desses *tiles* é muito menor. Essa estratégia, no entanto, possui algumas complexidades devido ao fato de não ser possível prever qual será o campo de visão do usuário em um instante no futuro, sendo necessária a aplicação de técnicas de predição de campo de visão com certas margens de segurança. A partir desse campo de visão expandido, pode-se priorizar quais *tiles* devem ser requisitados, sendo possível inclusive, uma abordagem preditiva de FOV.

Finalmente, além das adaptações dos algoritmos, pode-se tomar como estratégia a combinação dos diferentes algoritmos de maneira simultânea. Elaborou-se duas estratégias de combinação, uma mais conservadora e a outra mais arrojada. A mais conservadora combina os algoritmos de forma que sempre opta pelo de menor qualidade, dessa forma busca-se evitar travamento em detrimento de melhores qualidades. Já a estratégia arrojada faz o exato oposto, sempre escolhe o algoritmo que teve a melhor qualidade, presando então pela qualidade em detrimento da fluidez.

## 6 Conclusão

Atualmente existe diversos algoritmos de adaptação de taxa de transmissão. Com a expansão do uso e transmissão de vídeos 360°, pode-se notar que os algoritmos atuais serão de grande importância para esse novo cenário. Os atuais algoritmos passarão por adaptações para suportar vídeos 360°, elas serão principalmente voltados para campos de visão, dessa forma evitarão que a banda seja ocupada com conteúdo que não será assistido.

Em sintonia com os algoritmos de adaptação de taxa de transmissão, um grande aliado da transmissão de vídeos 360° será o protocolo QUIC. O fator chave que torna o QUIC tão importante é capacidade de criação de múltiplas *streams* com apenas uma conexão, dessa forma pode-se ter *streams* de mais e menos prioridades à depender do campo de visão.

## Referências

- [1] G. L. H. Rincão, W. G. da Cruz, D. M. Casas-Velasco and N. L. S. da Fonseca, *Impacto do tráfego background nas variantes BBR e CUBIC do protocolo TCP*
- [2] Fernandez, G. 360 Video streaming over QUIC. <https://github.com/gufernandez/aioquic-360-video-streaming>

- [3] K. Spiteri, R. Urgaonkar and R. K. Sitaraman, *BOLA: Near-Optimal Bitrate Adaptation for Online Videos*, in IEEE/ACM Transactions on Networking, vol. 28, no. 4, pp. 1698-1711, Aug. 2020, doi: 10.1109/TNET.2020.2996964.
- [4] Emily Marx, Francis Yan and Keith Winstein, *Implementing BOLA-BASIC on Puffer: Lessons for the use of SSIM in ABR logic*. <https://puffer.stanford.edu/bola/>
- [5] Fiscomp, *Métodos de Lyapunov*. [https://fiscomp.if.ufrgs.br/index.php/M%C3%A9todos\\_de\\_Lyapunov](https://fiscomp.if.ufrgs.br/index.php/M%C3%A9todos_de_Lyapunov)
- [6] Wikipedia, the free encyclopedia, *Lyapunov optimization* [https://en.wikipedia.org/wiki/Lyapunov\\_optimization](https://en.wikipedia.org/wiki/Lyapunov_optimization)