

# Videoconferência na Internet: Uma Caracterização de Qualidade de Serviço em Cenários Adversos

Arthur Böckmann Grossi<sup>1</sup>, Roberto Irajá Tavares da Costa Filho<sup>1,2</sup>,  
Luciano Paschoal Gaspar<sup>1</sup>

<sup>1</sup> Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

<sup>2</sup> Instituto Federal Sul-rio-grandense (IFSul)  
Rua General Balbão, 81 – 96.745-000 – Charqueadas – RS – Brasil

{abgrossi, roberto.costa, lpgaspar}@inf.ufrgs.br

**Abstract.** *With the increased use of video conferencing platforms in recent times, the quality of the provided service has become extremely important. The exchange of data, including audio and video, among participants in a video conference relies on a series of mechanisms and protocols that must function correctly to prevent communication degradation. This study examines the behavior of WebRTC-based video conferencing platforms when subjected to bandwidth restrictions. In this iteration, an analysis is conducted on the Elos and Google Meet platforms, using a cross-layer indicator-based approach for video conferencing service quality monitoring. The results indicate that the platforms employ distinct approaches to handle scenarios of limited network capacity. By deepening the understanding of these different adaptation strategies, this study contributes to the development of new quality adaptation strategies and provides support for research focused on evaluating QoE in video conferencing platforms.*

**Resumo.** *Com a intensificação no uso de plataformas de videoconferência nos últimos tempos, a qualidade do serviço provido tornou-se de extrema importância. A troca de dados, de áudio e vídeo, entre os participantes de uma videoconferência depende de uma série de mecanismos e protocolos, os quais precisam funcionar corretamente para que a comunicação não sofra degradações. Este estudo examina o comportamento de plataformas de videoconferência, baseadas em WebRTC, quando submetidas a restrições de largura de banda. Nesta iteração, realiza-se uma análise sobre as plataformas Elos e Google Meet, utilizando uma abordagem baseada em indicadores cross layer para monitoramento de qualidade de serviço de videoconferência. Os resultados obtidos indicam que as plataformas utilizam abordagens distintas para lidar com cenários de limitação em sua capacidade de rede. Ao aprofundar a compreensão dessas diferentes estratégias de adaptação, o presente trabalho fornece subsídio para o desenvolvimento de novas estratégias de adaptação de qualidade e para estudos voltados à avaliação de QoE em plataformas de videoconferência.*

## 1. Introdução

Nos últimos anos, tem-se observado uma transição gradual das atividades para o domínio digital. Cada vez mais, as pessoas estão executando suas tarefas cotidianas de maneira

remota, em regime de *home office*. Tanto conversas descontraídas quanto reuniões de trabalho, que antes aconteciam de forma presencial, migraram para o mundo digital fazendo uso de plataformas de videoconferência. Nesse contexto, algo que não possui um paralelo direto no cenário presencial tornou-se objeto de grande atenção: a *qualidade da experiência de videoconferência*. Câmeras travando, vozes picotadas e participantes sem responder são indícios de problemas muitas vezes oriundos de conexões de rede precárias, característica ainda *muito comum* em países como o Brasil.

Plataformas de videoconferência são um tipo de aplicação de tempo real que exige estabilidade de conexão bem como dos sistemas e serviços subjacentes. Até a menor das oscilações em algum fator pode causar uma degradação significativa na experiência do usuário e impactar a forma como é percebido e interage com os outros participantes. Exemplos desses fatores incluem a localização geográfica do usuário, a taxa de *download* e *upload* de pacotes, a latência da rede e a capacidade de processamento do dispositivo [da Costa Filho et al. 2016].

Diante do contexto mencionado, é *fundamental* que um provedor de serviço de videoconferência disponha de *caracterizações detalhadas, precisas e atualizadas sobre diferentes indicadores*, como os recém mencionados, que possam auxiliar na compreensão sobre o porquê determinados usuários estão tendo uma experiência satisfatória (ou não). Tal tem potencial, por exemplo, para subsidiar e justificar algum tipo de ação adaptativa dessas plataformas, até mesmo de maneira antecipada. Um exemplo é a troca para um *codec* de vídeo cujo *bitrate* médio seja mais adequado à largura de banda disponível. Adicionalmente, caracterizações podem oferecer suporte a aplicações que implementam o encaminhamento dinâmico de tráfego com base no desempenho da rede (a exemplo de [Tavares da Costa Filho et al. 2018]).

Apesar dos diversos estudos publicados no âmbito de aplicações de videoconferência e qualidade de experiência, os mesmos apresentam limitações importantes. Essas pesquisas têm utilizado valores de restrição de banda que não refletem a realidade de países em desenvolvimento, onde as redes operam com indicadores extremamente baixos [MacMillan et al. 2021, Ammar et al. 2016]. Além disso, carecem de uma análise específica das estratégias de adaptação e correção empregadas pelas plataformas para lidar com situações de degradação da rede.

Sendo assim, o presente estudo se propõe a comparar o desempenho de plataformas de videoconferência baseadas em WebRTC quando submetidas a condições de rede degradadas. Nesta iteração, realiza-se uma análise sobre as plataformas Elos (Mconf) e Google Meet, largamente empregadas no Brasil. Sem perda de generalidade, o aparato experimental pode ser expandido para outras plataformas. Os objetivos principais do estudo são avaliar e contrastar como essas plataformas, quando submetidas a redes de baixo desempenho, se adaptam e tentam manter a qualidade da experiência do usuário. Além da análise comparativa do comportamento das plataformas, o estudo também se destina a discutir as estratégias e abordagens adotadas por cada uma para lidar com tais situações adversas. Ao examinar essas estratégias, busca-se entender as soluções técnicas implementadas para melhorar a estabilidade e a qualidade de videoconferência em ambientes de rede com desempenho degradado.

É importante ressaltar que as interações *cross layer* dos protocolos e mecanis-

mos não apenas implicam a qualidade percebida dentro das plataformas de videoconferência, mas também desempenham um papel fundamental na definição da *Qualidade de Experiência (QoE)* para o usuário final [García et al. 2019]. Embora este trabalho não explore diretamente QoE como sua área final de estudo, os termos investigados, em última instância, serão responsáveis por definir a QoE. Portanto, a ênfase está na análise detalhada das interações das diferentes camadas de protocolos, implementação e aplicação, fornecendo subsídios para estudos futuros sobre como essas interações se refletem na percepção de qualidade por parte do usuário final.

O restante do artigo está organizado da seguinte forma. Na Seção 2, abordam-se as tecnologias e os fundamentos no âmbito de comunicação WebRTC, e apresentam-se e discutem-se os trabalhos relacionados. Na Seção 3, descreve-se a arquitetura montada para o processo de medição, assim como a seleção das métricas e o processo de implementação do aparato tecnológico. Na Seção 4, apresentam-se e analisam-se os resultados obtidos durante os experimentos, incluindo uma discussão sobre os principais *insights* extraídos dos experimentos. Por fim, na Seção 5, tece-se uma conclusão sobre o trabalho desenvolvido, contendo considerações finais e aspectos interessantes a serem abordados em trabalhos futuros.

## **2. Fundamentos e Estado da Arte**

Uma aplicação de videoconferência baseada em WebRTC fundamenta-se em uma série de mecanismos e protocolos subjacentes, dos quais é importante compreender aqueles que são essenciais para o seu funcionamento. Nesta seção, exploram-se os elementos-chave que sustentam essa tecnologia, desvendando sua arquitetura para comunicação em tempo real, transferência de dados e gestão de conexões. Além disso, ao final da seção, descrevem-se resumidamente e discutem-se os principais trabalhos relacionados.

### **2.1. Aplicações de Videoconferência via WebRTC**

Um dos pilares da troca de dados utilizando a tecnologia WebRTC é o protocolo RTP [Schulzrinne et al. 2003]. Esse protocolo é tipicamente implementado sobre o protocolo UDP na camada de transporte e, em conjunto com o protocolo RTCP, oferece ferramentas para lidar com desafios de transmissão de mídia em tempo real, como a compensação de atrasos e a detecção de perdas de pacotes. Apesar de oferecer essas ferramentas, o protocolo RTP não estipula limites para latência ou confiabilidade; sendo assim, fica a cargo da aplicação utilizá-las ou não para corrigir problemas intercorrentes.

No protocolo RTP (Real-time Transport Protocol), as sessões desempenham um papel fundamental na organização e transmissão de mídia em tempo real. Uma sessão é definida como um grupo de participantes que se comunicam por meio desse protocolo, sendo projetada para transportar um único tipo de mídia, como áudio ou vídeo. Essencialmente, cada sessão é identificada por um número de identificação de sessão (SSRC - Synchronization Source) exclusivo para cada fonte de mídia, permitindo a distinção e a correta interpretação dos fluxos de dados entre os participantes. Durante uma sessão RTP, os pacotes de mídia são enviados da fonte para os outros participantes, cada um contendo um cabeçalho com informações cruciais, como número de sequência, marcação de eventos e *timestamps*, facilitando a identificação e o processamento adequado da mídia transmitida. Em paralelo ao protocolo RTP, opera o protocolo RTCP (Real-time Transport Control Protocol), que desempenha um papel complementar crucial. Enquanto o

RTP se concentra na transmissão eficiente da mídia, o RTCP é responsável pelo *feedback* e monitoramento da qualidade da comunicação em tempo real. Periodicamente, os participantes enviam pacotes RTCP contendo estatísticas, como a quantidade de pacotes perdidos, atrasos médios e informações sobre a qualidade da conexão. Essa troca de informações permite ajustes dinâmicos na transmissão para otimizar a experiência do usuário, adaptando-se a condições variáveis da rede durante a sessão RTP. Dessa forma, a combinação dos protocolos RTP e RTCP permite uma comunicação contínua e eficiente em ambientes de transmissão em tempo real.

O estabelecimento de uma sessão RTP acontece na etapa de conexão do protocolo WebRTC. Durante essa fase, os agentes WebRTC passam por uma sequência de processos para permitir a comunicação direta entre eles. A sinalização é o primeiro passo, onde os agentes trocam informações por meio do protocolo Session Description Protocol (SDP)[Begen et al. 2021]. Nesse ponto, detalhes como endereços IP, portas, formatos de mídia e *codecs* são compartilhados para iniciar a comunicação. Em seguida, inicia-se a etapa de conexão propriamente dita, onde utiliza-se o protocolo Interactive Connectivity Establishment (ICE)[Keränen et al. 2018] para superar os desafios de conectividade, contornando *firewalls* e NATs para descobrir os IPs públicos dos agentes. Tal é realizado por meio de testes, que identificam os melhores caminhos para estabelecer a conexão direta ou indireta entre os pares. Por fim, a proteção da conexão é garantida pelo Datagram Transport Layer Security (DTLS) [Rescorla et al. 2022] para estabelecer uma sessão segura, utilizando o protocolo Secure Real-time Transport Protocol (SRTP) [Carrara et al. 2004] para a transferência criptografada de mídia.

Após a conclusão bem-sucedida das fases de estabelecimento de uma conexão WebRTC, a troca de dados se inicia. Em videoconferências, onde há vários participantes trocando dados de voz e vídeo simultaneamente, é crucial organizar a comunicação de forma a aumentar a eficiência da troca de dados. No ecossistema WebRTC, diversas topologias de comunicação são possíveis para a interação entre os participantes. Uma abordagem comum é a topologia de malha completa, na qual cada integrante da videoconferência estabelece conexões diretas com todos os outros participantes. Embora essa configuração minimize a latência, ela leva a um aumento significativo da quantidade de conexões, o que pode sobrecarregar os clientes em termos de processamento e largura de banda, especialmente em grandes grupos. Uma alternativa é a utilização da Multipoint Conferencing Unit (MCU), uma topologia cliente-servidor. Nesse cenário, cada agente WebRTC se conecta a um servidor central, que recebe, re-codifica e redistribui as mídias, consolidando-as em um único fluxo. Embora essa abordagem acrescente uma certa latência devido à presença do servidor central, ela alivia os clientes da carga pesada de processamento e do uso excessivo de banda. Uma terceira opção é a Selective Forwarding Unit (SFU), também baseada em um modelo cliente-servidor. Nessa configuração, o servidor central não agrega todos os fluxos de mídia em um único, mas os redistribui como fluxos separados. Isso proporciona flexibilidade ao servidor para ajustar os fluxos de mídia, como na disposição de *layouts*, sem a necessidade de reprocessar toda a mídia. Cada uma dessas topologias atende a diferentes requisitos e cenários, permitindo a adaptação a diversas situações de aplicação de videoconferência.

## 2.2. Trabalhos Relacionados

No passado recente, foram publicados artigos que buscam estudar diferentes aspectos relacionados a aplicações de videoconferência e qualidade. A seguir, descrevem-se dois trabalhos identificados como mais representativos.

O trabalho do autor Doreid Ammar [Ammar et al. 2016], publicado em 2016, explora as estatísticas de uma sessão WebRTC a fim de identificar a potencial relevância dessas informações em indicar problemas na qualidade da experiência dos usuários de uma videoconferência. Para isso, o autor realizou uma série de experimentos em um ambiente de videoconferência com dois usuários, utilizando a plataforma *appear.in*. Durante os testes, foram simuladas diferentes condições de qualidade de conexão em ambos os participantes e, em seguida, analisou-se como essas condições refletiam nas estatísticas geradas pelo protocolo WebRTC. Nesse trabalho ficou claro que existem diversas métricas oferecidas pelo protocolo WebRTC que permitem estimar a qualidade da experiência do usuário.

No ano de 2021, o autor Kyle MacMillan publicou o trabalho [MacMillan et al. 2021], que tem como objetivo entender a utilização de rede por aplicações de videoconferência. O estudo analisa a variação das métricas de desempenho sob diferentes capacidades de vazão, a resposta das aplicações de videoconferência a interrupções na comunicação e diferentes modalidades de uso, bem como o comportamento dessas aplicações na presença de outros fluxos. O trabalho conclui que as três aplicações de videoconferência estudadas variam significativamente nos experimentos aplicados devido aos seus diferentes mecanismos de controle de congestionamento e de transporte, assim como pelas estratégias de codificação e infraestrutura utilizadas. Outro ponto demonstrado é que a utilização de rede por essas aplicações pode variar significativamente, a depender de configurações de *layout* aplicadas. Por exemplo, há casos em que conferências com menos participantes utilizam uma maior largura de banda do que conferências com um maior número de participantes para a mesma plataforma.

Como se observa na literatura, é muito promissora a utilização de indicadores quantitativos para caracterizar o funcionamento de aplicações de videoconferência. A monitoração desses indicadores permite uma análise ampla do comportamento dessas aplicações e sua qualidade em relação a diferentes aspectos, desde restrições em maior ou menor escala até o uso de diferentes *codecs*. A partir desses estudos, é possível obter informações relevantes sobre o contexto da videoconferência e, assim, entender melhor o comportamento dessas aplicações.

Inspirado nessas pesquisas, o presente trabalho propõe uma análise comparativa entre as plataformas de videoconferência: Elos, utilizada por mais de 700 mil usuários<sup>1</sup>, que ainda carece de estudos específicos, e a plataforma Google Meet. O objetivo é realizar uma análise mais detalhada do tráfego de aplicações de videoconferência reais, buscando compreender seus perfis de uso e entender como essas aplicações se comportam em situações de restrições de rede. Para tal, simularam-se cenários que estão mais próximos da realidade de países, como o Brasil, onde a capacidade de Internet disponível à população tende a ser limitada. Ressalta-se que os resultados apresentados para o Elos

---

<sup>1</sup><https://blog.elos.vc/como-a-pearson-adaptou-se-aos-desafios-da-pandemia-atraves-do-ensino-a-distancia/>

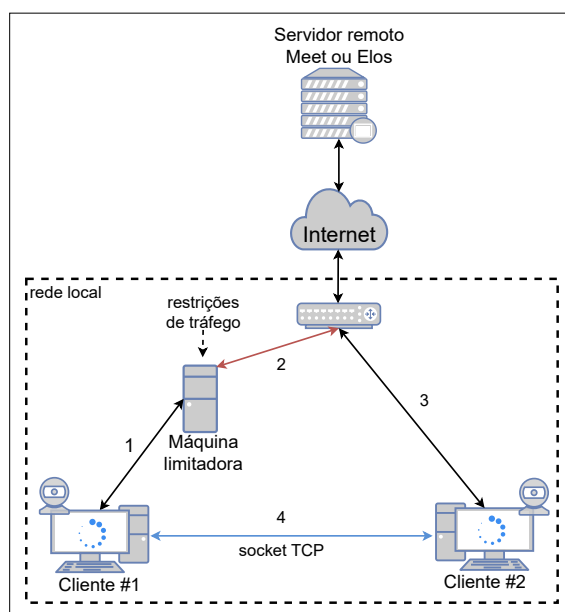
são igualmente válidos e se aplicam para *todos* os projetos baseados em BigBlueButton<sup>2</sup>, permitindo uma compreensão mais clara do desempenho dessas plataformas em diferentes contextos e trazendo insumos relevantes para o desenvolvimento de mecanismos que promovam uma melhor qualidade de experiência.

### 3. Medições de Qualidade em Plataformas WebRTC

A fim de viabilizar esse estudo, foi necessário o desenvolvimento de três pontos: (i) a elaboração de uma arquitetura de medições que permita a realização de medidas de maneira controlada, sem que sofram interferências de fontes externas; (ii) o levantamento de indicadores adequados, os quais permitam atestar o efeito de degradações de rede na qualidade de experiência do usuário de aplicações de videoconferência; e (iii) a implementação de ferramentas de software auxiliares que permitam a automatização do processo de experimentação. Na sequência desta seção, apresenta-se cada um desses pontos em detalhe.

#### 3.1. Arquitetura de Medição

Para poder realizar um conjunto extensivo de experimentos que envolvam aplicações de videoconferência, sob diferentes níveis de degradação de rede, foi considerado um ambiente experimental composto por duas máquinas cliente e uma máquina limitadora, conforme ilustra a Figura 1. Ambas máquinas cliente executam *scripts* separados e se comunicam entre si por meio de um *socket* TCP, representado pela conexão 4. Tal é necessário para a sincronização da entrada e saída nas salas de videoconferência durante a execução dos experimentos. Além disso, as máquinas cliente possuem recursos suficientes para executar as aplicações de videoconferência e se comunicam, através da Internet, com os servidores dessas plataformas.



**Figura 1. Ambiente experimental**

Sobre essa infraestrutura, que está em sintonia com a utilizada em trabalhos anteriores[da Costa Filho et al. 2016, Ammar et al. 2016] e de forma especial no trabalho

<sup>2</sup><https://github.com/bigbluebutton/bigbluebutton>

[MacMillan et al. 2021], a ideia é criar um mecanismo que permita, de forma controlada, gerar degradação em parâmetros de funcionamento do canal de comunicação e, assim, observar os efeitos na experiência de uso da plataforma. Para aplicar as limitações no canal de comunicação, foi utilizada a ferramenta Linux Traffic Control (TC) sendo executada em uma máquina isolada. Tal tem como objetivo evitar que possíveis ruídos causados pelo uso de recursos para aplicação das restrições afetem as máquinas cliente durante a execução dos *scripts*. A máquina limitadora liga-se diretamente à primeira máquina cliente pela conexão 1 e conecta-se com a Internet por um canal de comunicação representado pela conexão 2, onde são aplicadas as restrições. A segunda máquina cliente se conecta diretamente à Internet, sem restrições, pela conexão 3.

### 3.2. Métricas de Interesse

Tendo em vista a vasta quantidade de indicadores sobre aspectos de qualidade da comunicação que a API WebRTC oferece, assim como a facilidade na sua captura, acabou-se utilizando esses indicadores como a fonte mais importante de informação sobre uma videoconferência. Como essa API dá acesso a mais de 200 métricas, sendo elas muito diversas, optou-se por fazer uma triagem sobre esse conjunto, a fim de identificar as mais relevantes para este trabalho.

A Tabela 1 apresenta as métricas que foram selecionadas. A análise desses indicadores é um processo complexo por si só e, por esse motivo, selecionou-se um conjunto reduzido de métricas, a fim de permitir a realização de um exercício de qualidade em compreender como elas se relacionam. O conjunto final contém métricas que satisfazem a dois critérios: (i) possuem relação direta com a experiência de uso e (ii) são suportadas pelos três principais navegadores (Safari, Mozilla Firefox e Google Chrome). As métricas derivadas que foram utilizadas neste trabalho, assim como suas fórmulas, são apresentadas na Tabela 2. Métricas derivadas são indicadores que são calculados pelos próprios navegadores, com base em outras métricas base.

Grupo	Métrica	Descrição
OutboundRTP	targetBitrate	Taxa de bits alvo do codificador
OutboundRTP	bytesSent	Número total de bytes enviados para esse SSRC
OutboundRTP	frameWidth	Largura do último <i>frame</i> codificado
OutboundRTP	frameHeight	Altura do último <i>frame</i> codificado
OutboundRTP	framesPerSecond	Número de quadros codificados no último segundo
OutboundRTP	qpSum	Soma dos valores QP dos quadros codificados
OutboundRTP	framesEncoded	Número de quadros codificados

**Tabela 1. Métricas observadas ao longo dos experimentos**

Grupo	Métrica	Fórmula
OutboundRTP	[bytesSent_in_bits/s]	$8 \cdot \frac{\text{currReport.BytesSent} - \text{prevReport.BytesSent}}{\text{currReport.Time} - \text{prevReport.Time}}$
OutboundRTP	[qpSum/framesEncoded]	$\frac{\text{qpSum}}{\text{framesEncoded}}$

**Tabela 2. Métricas derivadas calculadas no navegador**

### 3.3. Implementação

Dada a arquitetura de medições e as métricas de interesse apresentadas na Subseção 3.2, parte-se para a implementação e a instanciação de um ambiente de experimentos que permita a realização de um conjunto de observações. Foram desenvolvidos dois grupos de *scripts*. O primeiro grupo é responsável pela execução dos experimentos, onde toda a interação com a interface das plataformas Elos e Google Meet foi automatizada,

assim como a aplicação das restrições na rede, o que possibilita a repetição dos experimentos sem demandar demasiado esforço manual. O segundo *script* é responsável pela interpretação e geração dos gráficos com base nos dados coletados durante a execução do primeiro conjunto de *scripts*.

Para a automatização dos experimentos, utilizou-se como base o *script* Python<sup>3</sup> desenvolvido no trabalho [MacMillan et al. 2021], que, em certa medida, possui compatibilidade com as plataformas Google Meet, Zoom e Microsoft Teams, mas não para o Elos. Com o intuito de facilitar futuras adições de funcionalidades e melhorar o funcionamento do código, realizou-se um *refactor* significativo nesse *script*<sup>4</sup>, tornando-o mais modular e facilmente extensível para suportar outras plataformas além das já mencionadas. Além disso, expandiu-se o fluxo do *script* para suportar a interação simultânea de dois clientes, introduzindo sincronização entre eles. Também incluiu-se a capacidade de gravação de tela, enriquecendo significativamente o escopo do experimento.

Para compatibilização com a plataforma Elos, mapearam-se e implementaram-se todas as interações necessárias para o ingresso em uma videoconferência, assim como compartilhamento de câmera e saída de uma videoconferência no Elos. Para tal, o *script* utiliza duas bibliotecas que trabalham em conjunto, a PyAutoGui, que executa cliques e controla o teclado, e a GuiBot, que identifica elementos na tela por meio de visão computacional e inteligência artificial, permitindo que elementos sejam identificados somente pela captura de tela. O fluxo dos *scripts* é descrito a seguir.

No início de cada experimento, o *script* faz a leitura de um arquivo texto, onde estão definidas as restrições de rede a serem aplicadas. Para cada restrição de rede, por padrão, são repetidas cinco rodadas do experimento, visando a uma maior base estatística, sendo que esse número deve ser ajustado em função da variabilidade de desempenho do ambiente em questão. Uma vez que uma rodada inicia, um conjunto de restrições é aplicado por meio de uma conexão SSH com a máquina limitadora, e, com o auxílio da biblioteca ffmpeg e o módulo v4l2loopback, é emulado um dispositivo de câmera reproduzindo em ciclo um vídeo predeterminado do tipo *talking head*. Então, depois de entrar na conferência e habilitar a câmera, inicia-se uma captura de pacotes que dura um tempo pré-configurado. Ao final desse tempo, salva-se o relatório fornecido pela página interna<sup>5</sup> do navegador Google Chrome, onde consta uma série de informações sobre a sessão WebRTC.

Após a realização de um experimento, os dados coletados são copiados para um armazenamento em nuvem para que sejam acessados por um segundo *script* Python que executa no ambiente Google Colab. Nele, foi implementado um analisador de arquivos de *dump* WebRTC, que extrai as informações e as armazena em uma estrutura de dados em memória, para que, com auxílio da biblioteca matplotlib, os gráficos sejam plotados. Esse *script* foi pensado para que seja possível alterar, de forma simples, os arquivos a serem analisados, assim como as métricas que irão compor os gráficos.

---

<sup>3</sup><https://github.com/kyle-macmillan/vca-imc-21>

<sup>4</sup><https://github.com/ArthurK12/vca-imc-21/tree/elos-benchmark>

<sup>5</sup><chrome://webrtc-internals>



## 4. Resultados

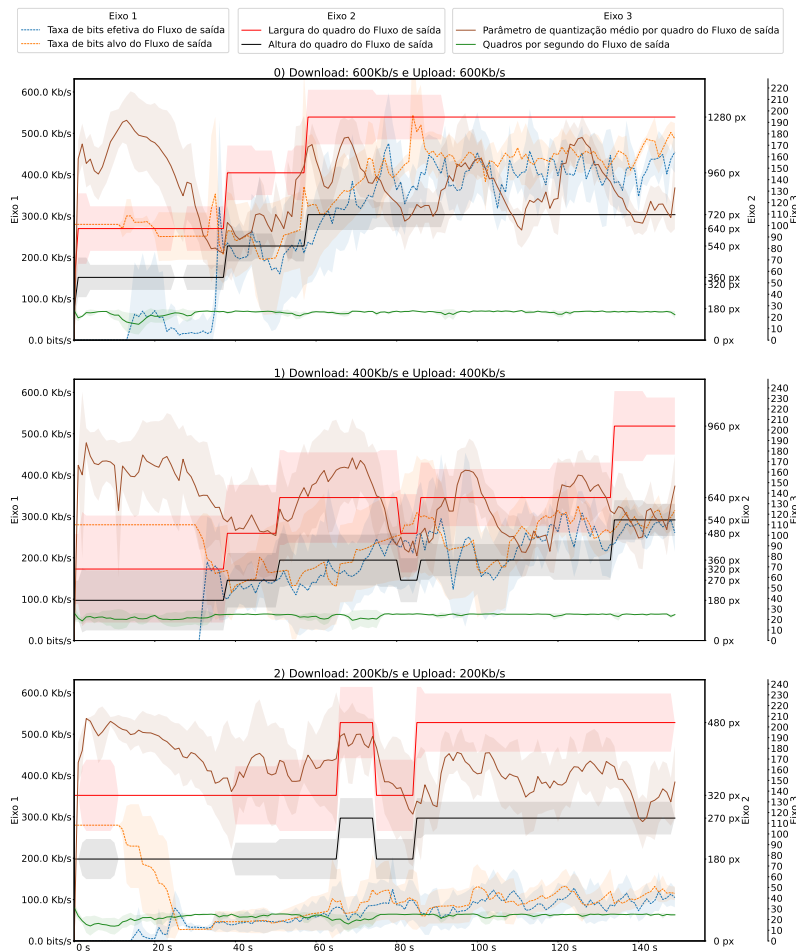
Nesta seção, são apresentados os resultados obtidos no desenvolvimento deste trabalho. Com o intuito de enriquecer a análise, os resultados foram divididos em duas subseções. Primeiramente, apresentam-se os resultados e após, uma discussão com os principais *insights* alcançados com os experimentos.

### 4.1. Entendendo o Comportamento de um Vídeo sob Restrições de Banda

As plataformas Elos e Google Meet são amplamente utilizadas por uma grande variedade de usuários em diferentes locais, com diferentes operadoras de Internet, bem como distintos planos e capacidades. Diante desse cenário, compreender o comportamento de um vídeo sob vários níveis de capacidade de rede é extremamente importante para tomar conhecimento do impacto desse tipo de limitação de rede na qualidade da experiência dos usuários e entender as estratégias adotadas por cada plataforma para lidar com tais situações. Tendo em vista esse aspecto, foram realizados dois experimentos, um para cada plataforma, com três níveis diferentes de capacidade de rede (200 Kbps, 400 Kbps e 600 Kbps, todas simétricas, *download* e *upload*), escolhidos levando em conta a especificação dos requisitos das plataformas. Foram realizadas cinco rodadas em cada nível, cada uma com 150 segundos de duração. O experimento consiste de dois usuários em uma videoconferência, cada um deles compartilhando câmera na qualidade máxima. As Figuras 2 e 3 ilustram o comportamento das plataformas frente a esses diferentes níveis de restrição. Em cada gráfico, são mostradas as curvas da média da taxa efetiva de bits por segundo, da taxa de bits por segundo alvo, do parâmetro de quantização médio e dos quadros por segundo. As curvas da largura e da altura do quadro denotam a representação da moda dessas medidas. O sombreamento representa o desvio padrão em todas as curvas.

Ao analisar os gráficos da Figura 2, percebe-se que a plataforma Google Meet, conforme esperado, alcança uma resolução maior à medida que as restrições de banda diminuem. Na restrição de 200 Kbps, a resolução máxima atingida é de 480x270 *pixels*, na restrição de 400 Kbps a resolução máxima é de 960x540 *pixels*, e na restrição de 600 Kbps atinge 1280x720 *pixels*. Em todos os três cenários, há pouca oscilação na taxa de quadros por segundo, que se mantém aproximadamente estável no valor de 30. É importante frisar que nos três cenários restritos, o envio do vídeo inicia na mesma resolução de 320x180 *pixels* com a taxa de bits por segundo alvo no patamar de 300 Kbps, o que pode indicar uma estratégia conservadora por parte da plataforma para não extrapolar o limite da rede. Um fato curioso é que a taxa de bits efetiva se mantém baixa até, aproximadamente, os 20 segundos dos experimentos, quando, então, começa a se elevar. Isso indica que pode haver algum tipo de mecanismo onde o envio do vídeo só começa após o ajuste da taxa de bits alvo, para que essa esteja mais próxima do valor “real” da rede.

A Figura 3 ilustra o comportamento da plataforma Elos, apresentando gráficos para apenas dois cenários restritos (400 Kbps e 600 Kbps). A razão para isso é que nos experimentos com uma restrição de banda de 200 Kbps, o envio e o recebimento de vídeos não foram mantidos consistentemente ao longo dos 150 segundos. Consequentemente, não havia dados suficientes para gerar os gráficos correspondentes. Essa situação sugere que a plataforma Elos aplica uma estratégia de não reconectar automaticamente os vídeos dos participantes quando a rede não atende a um requisito mínimo. Ao analisar os gráficos

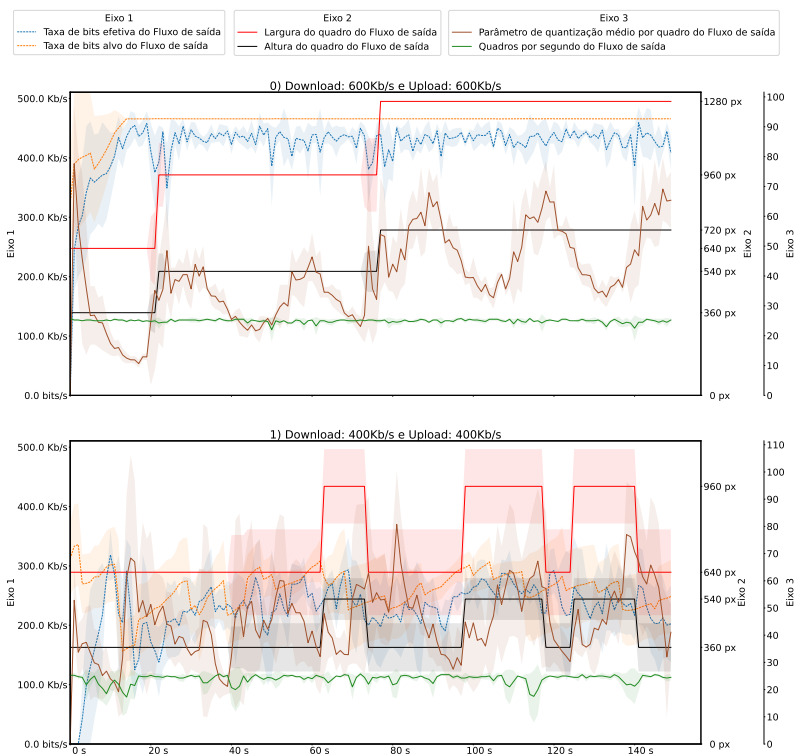


**Figura 2. Desempenho do Google Meet frente às restrições de 600, 400 e 200 Kbps**

para as restrições de 600 Kbps e 400 Kbps, algumas diferenças em relação à plataforma Google Meet tornam-se evidentes. Primeiramente, no Elos, o parâmetro de quantização assume valores menores, com o máximo sendo 100, enquanto no Meet o valor máximo é de 240. Em segundo lugar, no Elos as curvas da taxa de bits alvo e da taxa de bits efetiva aproximam-se logo no início do experimento e têm um crescimento acelerado, diferente do comportamento do vídeo no Google Meet. Em terceiro lugar, observa-se o comportamento da curva da taxa de bits alvo no cenário com restrição de 600 Kbps. No Elos, a curva estabiliza-se em 600 Kbps, indicando que a plataforma está “satisfeita” com essa largura de banda, pois o perfil de câmera aplicado consome aproximadamente 500 Kbps. Por outro lado, no Google Meet, sob a mesma restrição, a curva da taxa de bits alvo apresenta um padrão de “dente de serra”, sugerindo que a plataforma está testando a rede para determinar se pode utilizar mais largura de banda, possivelmente para aumentar os parâmetros do vídeo.

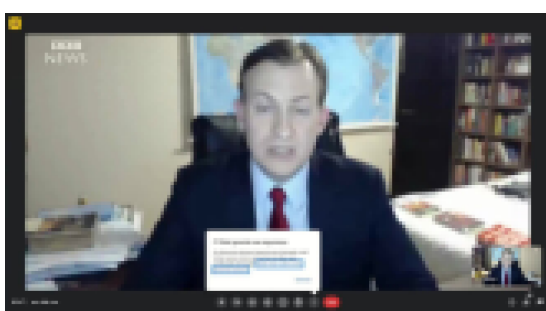
## 4.2. Discussão

As plataformas de videoconferência enfrentam muitos desafios na busca por maximizar a Qualidade de Experiência (QoE) dos usuários, desde a gestão da taxa de bits enviados de um vídeo em cenários adversos até a tomada de decisões delicadas durante degradações de rede, tais como a reconexão automática de câmeras. Esses desafios complexos requerem um equilíbrio delicado entre a adaptação dinâmica às condições variáveis da rede e a

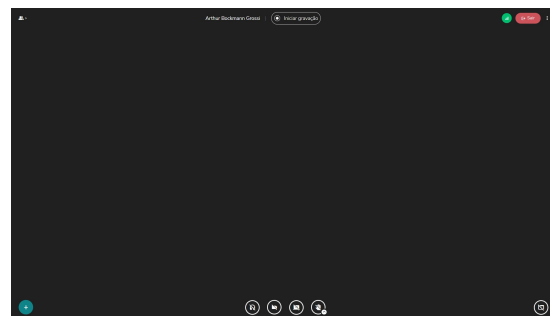


**Figura 3. Desempenho do Elos frente às restrições de 600 e 400 Kbps**

entrega consistente de uma experiência de alta qualidade aos usuários. Cada plataforma de videoconferência aborda esses problemas de formas diferentes, dependendo de detalhes da infraestrutura, proposta da aplicação e quais mecanismos adicionais implementados. A seguir discutem-se os principais *insights* do estudo realizado.



(a) Google Meet



(b) Elos

**Figura 4. Comparação entre a experiência do usuário na plataforma Google Meet e Elos sob restrição de 200Kbps**

**Conexão e reconexão de vídeo.** Quando um participante de uma videoconferência está passando por problemas em sua conexão, o recebimento dos vídeos e da voz dos outros participantes, assim como o envio de vídeo e voz, podem ser afetados. Em casos extremos, em que a rede sofre uma restrição muito grande em sua largura de banda e não possui capacidade suficiente para manter o canal dos vídeos “aberto”, existem algumas abordagens possíveis para enfrentar esse problema. A estratégia adotada pela plataforma Elos é a seguinte: após alguns segundos tentando a reconexão dos vídeos sem sucesso, a reconexão automática é cessada, conforme destacado na Figura 4(b). Apesar de não res-

taurar o compartilhamento de vídeo automaticamente após a estabilização da rede, essa estratégia parece coesa, pois, ao deixar de insistir na reconexão das câmeras, a plataforma “assume” que a rede do usuário, no momento, não é capaz de suportar tal fluxo de dados, priorizando a troca de dados de áudio. O Google Meet, no mesmo cenário, reduz ao máximo a resolução do vídeo para se adequar à capacidade momentânea da rede, ainda que essa estratégia se traduza em exibir um vídeo em tamanho grande com resolução extremamente baixa, como ilustra a Figura 4(a).

**Prioridades de fluxos.** Outro desafio está na evolução da taxa de bits do vídeo enviado para otimizar o uso da rede. Conforme apresentado nos gráficos anteriores, há uma diferença significativa no comportamento das duas plataformas em relação às curvas da taxa de bits enviados e da taxa de bits alvo (estimativa). O Google Meet e o Elos têm diferenças cruciais entre si que, em parte, são explicadas pela forma como o áudio e o vídeo são enviados nas duas plataformas. No Google Meet, áudio e vídeo são enviados no mesmo transporte; isso significa que é criada uma única sessão RTP para enviar o áudio e o vídeo do participante. Essa estratégia permite a designação de prioridade entre áudio e vídeo, o que é essencial em situações de instabilidade na rede. Quando a rede enfrenta dificuldades, a taxa de envio do fluxo com menor prioridade pode ser reduzida antes que seja necessário diminuir a do fluxo com maior prioridade. Por outro lado, no Elos, ao enviar dados em duas sessões RTP separadas, essas sessões competem entre si pelo uso da rede, o que pode levar a desafios adicionais de gerenciamento de largura de banda e, conseqüentemente, a uma adaptação menos eficiente às condições variáveis da rede.

**Mecanismos de transmissão.** Além das diferenças na forma como o áudio e o vídeo são tratados, as variações na evolução da taxa de bits por segundo entre o Google Meet e o Elos podem ser atribuídas à implementação de mecanismos que visam a aumentar a eficiência do uso da rede. Um desses recursos é o *simulcast*, adotado pelo Google Meet. O *simulcast* permite a transmissão de um mesmo vídeo em diferentes resoluções ou qualidades codificadas simultaneamente em uma única fonte. Durante a transmissão, o servidor pode selecionar qual dessas versões enviar para cada destinatário, com base na capacidade de rede reportada pelo receptor. Esse mecanismo permite uma adaptação mais dinâmica, pois ajusta a resolução do vídeo, e conseqüentemente a taxa de bits, adaptando a mídia de acordo com a largura de banda disponível para cada destinatário. Enquanto essa estratégia oferece vantagens na otimização da largura de banda, é crucial notar que ela demanda mais recursos computacionais no cliente que envia o vídeo, pois requer a codificação do vídeo em múltiplas versões para oferecer opções adaptativas. Por outro lado, no servidor, o aumento dos recursos está associado à gestão e indicação das diferentes versões do vídeo a serem enviadas aos clientes.

**Estimativa de largura de banda.** Elos e Google Meet possuem uma diferença notável na abordagem inicial da estimativa de banda. No Google Meet, a estimativa inicial segue o padrão da *libwebrtc*, fixada em 300kbps<sup>6</sup>. Isso explica por que o Google Meet sempre inicia com essa estimativa, independentemente das restrições de rede, sendo ajustada ao longo do experimento. Por outro lado, o Elos utiliza uma estratégia personalizada, determinando a estimativa inicial com base na taxa de bits necessária para o perfil de câmera

---

<sup>6</sup>[https://source.chromium.org/chromium/chromium/src/+main:third\\_party/webrtc/api/transport/bitratesettings.h;l=45;drc=0e61fdd27c3371b75dba81252511ce0becfa78a0](https://source.chromium.org/chromium/chromium/src/+main:third_party/webrtc/api/transport/bitratesettings.h;l=45;drc=0e61fdd27c3371b75dba81252511ce0becfa78a0)

usado. Sempre utilizando o perfil de câmera “Alta Definição” (500kbps) nos experimentos, a estimativa de banda tende a esse valor no início, mostrando-se mais eficaz em redes com restrições mais severas que a taxa de bits necessária para o perfil de câmera. Em tais casos, os usuários desfrutam de vídeos de melhor qualidade por mais tempo. Essa estratégia customizada introduz um parâmetro adicional ( $b=AS[\textit{bitrate\_perfil\_camera}]$ ) ao SDP (Session Description Protocol) durante a negociação, indicando a estimativa de banda inicial desejada desde o início da comunicação, alinhando-a com a necessidade específica do perfil de câmera. Esse método influencia a adaptação inicial da transmissão de acordo com as capacidades de largura de banda disponíveis.

## 5. Conclusão

Nos últimos anos, houve uma mudança significativa na maneira como as pessoas se comunicam, trabalham e estudam. As plataformas de videoconferência tornaram-se uma escolha popular para a comunicação global. Como resultado, houve um aumento significativo no foco na qualidade da experiência do usuário em videoconferências. É crucial compreender como essas plataformas funcionam quando os usuários enfrentam condições degradadas na rede, para que seja possível desenvolver e aprimorar os mecanismos de adaptação.

Neste trabalho, criou-se um arcabouço a fim de caracterizar e comparar o comportamento de plataformas de videoconferência, e o instanciou-se para analisar o Elos e o Google Meet sob diferentes condições de rede. Ao simular ambientes com degradações nos parâmetros de *download* e *upload* da rede, constatou-se que as plataformas de videoconferência são capazes de se adaptar a condições severas de rede de forma eficiente. Além disso, ficou claro que cada plataforma possui uma estratégia e uma forma diferente de lidar com essas circunstâncias, sendo essas intimamente ligadas a outros aspectos como infraestrutura do provedor, mecanismos de controle de congestionamento, mecanismos de transmissão de vídeo, topologia, entre outros. Os resultados obtidos indicam que a interação eficiente entre os múltiplos mecanismos, implementações e protocolos, no contexto de videoconferência utilizando o protocolo WebRTC, é bem-sucedida. No entanto, é importante destacar que a interface de múltiplas implementações acrescenta uma complexidade elevada ao funcionamento desse protocolo WebRTC e que, não raramente, as mesmas extrapolam os padrões, resultando em possíveis comportamentos peculiares e dificuldades operacionais.

Dessa forma, este trabalho contribui para o desenvolvimento de estudos que visem estimar a QoE, baseando-se em uma compreensão mais refinada dos impactos da Internet em QoS. No entanto, há ainda diversos caminhos promissores a serem explorados em trabalhos futuros. Primeiramente, destaca-se a necessidade de aprofundar o estudo de QoE, considerando variações mais específicas e detalhadas em diferentes condições de rede. Essa abordagem mais detalhada pode permitir uma análise mais precisa dos fatores que influenciam diretamente a percepção do usuário em ambientes de videoconferência, contribuindo para estratégias mais eficazes de melhoria da experiência.

Outro ponto de destaque para trabalhos futuros é a exploração de experimentos que envolvam mobilidade. Essa abordagem implica a consideração de variações nas condições de rede em curtos espaços de tempo, simulando situações em que os participantes estão em movimento. Esses experimentos podem revelar *insights* valiosos sobre

como as plataformas de videoconferência se comportam diante de cenários dinâmicos, permitindo uma compreensão mais abrangente e realista do desempenho dessas plataformas. Em resumo, o presente trabalho estabelece uma base para investigações futuras em QoE e a exploração de condições mais dinâmicas, agregando maior complexidade aos experimentos e, conseqüentemente, enriquecendo as conclusões e recomendações para aprimoramento das plataformas de videoconferência.

## Agradecimentos

O presente trabalho foi desenvolvido com o apoio do projeto UFRGS-Netmetric e do projeto FAPESP 2020/05183-0. Agradecemos, também, à equipe do Elos pelo apoio e pelos *insights* para o aprimoramento do trabalho.

## Referências

- [Ammar et al. 2016] Ammar, D., De Moor, K., Xie, M., Fiedler, M., and Heegaard, P. (2016). Video qoe killer and performance statistics in webrtc-based video communication. In *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, pages 429–436.
- [Begen et al. 2021] Begen, A. C., Kyzivat, P., Perkins, C., and Handley, M. J. (2021). SDP: Session Description Protocol. RFC 8866.
- [Carrara et al. 2004] Carrara, E., Norrman, K., McGrew, D., Naslund, M., and Baugher, M. (2004). The Secure Real-time Transport Protocol (SRTP). RFC 3711.
- [da Costa Filho et al. 2016] da Costa Filho, R. I. T., Lautenschläger, W., Lazzari, H., Roesler, V., and Gaspary, L. P. (2016). A rede tem a resposta: Um modelo escalável para predição integrada de qualidade de vídeo e qoe em redes ip. In *Proceedings of the XXXIV Brazilian Symposium on Computer Networks and Distributed Systems (SBRC), Salvador, Bahia. (Honorable Mention Award-Top 3)*.
- [García et al. 2019] García, B., Gallego, M., Gortázar, F., and Bertolino, A. (2019). Understanding and estimating quality of experience in webrtc applications. *Computing*, 101(11):1585–1607.
- [Keränen et al. 2018] Keränen, A., Holmberg, C., and Rosenberg, J. (2018). Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal. RFC 8445.
- [MacMillan et al. 2021] MacMillan, K., Mangla, T., Saxon, J., and Feamster, N. (2021). Measuring the performance and network utilization of popular video conferencing applications. In *Proceedings of the 21st ACM Internet Measurement Conference, IMC '21*, page 229–244, New York, NY, USA. Association for Computing Machinery.
- [Rescorla et al. 2022] Rescorla, E., Tschofenig, H., and Modadugu, N. (2022). The Datagram Transport Layer Security (DTLS) Protocol Version 1.3. RFC 9147.
- [Schulzrinne et al. 2003] Schulzrinne, H., Casner, S. L., Frederick, R., and Jacobson, V. (2003). RTP: A Transport Protocol for Real-Time Applications. RFC 3550.
- [Tavares da Costa Filho et al. 2018] Tavares da Costa Filho, R. I., Lautenschläger, W., Kagami, N., Caggiani Luizelli, M., Roesler, V., and Paschoal Gaspary, L. (2018). Scalable qoe-aware path selection in sdn-based mobile networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 989–997.