

Ambiente Programável para Transmissão e Análise de Qualidade de Experiência de Vídeo Volumétrico

Alan Teixeira da Silva¹, Fabricio R. Cesen², Rafael Clerici¹, Christian E. Rothenberg¹, Eduardo Cerqueira³, Vanessa Testoni¹

¹Universidade Estadual de Campinas (UNICAMP)
Campinas – Brazil

²Telefonica Research
Barcelona – Spain

³Universidade Federal do Pará (UFPA)
Belém – Brazil

{a265560, r273034}@dac.unicamp.br, {chesteve, vtestoni}@unicamp.br

fabricio.rodriguezcesen@telefonica.com

cerqueira@ufpa.br

Abstract. *The expansion of 5G networks and the growth of data traffic has driven the development of innovative applications, particularly Holographic-Type Communications (HTC), paving the way for 6G networks. This work presents a comparative study of Quality of Experience (QoE) in volumetric video transmission within immersive virtual reality environments, using TCP-HTTP and WebRTC protocols. Main contributions include the development of a programmable hardware-based emulation environment based on the P4/Tofino technologies of the P4 Programmable Patch Panel (P7) tool, evaluation of network metrics impact on QoE during volumetric video transmission in point cloud format rendered on a real Head Mounted Display (HMD) device (Meta Quest 3), implementation and comparative analysis between different transport protocols.*

Resumo. *A expansão das redes 5G e o crescimento do tráfego de dados impulsionaram o desenvolvimento de aplicações inovadoras, particularmente as Comunicações do Tipo Holográfico (HTC), que está pavimentando o caminho para redes 6G. Este trabalho apresenta um estudo comparativo da Qualidade de Experiência (QoE) na transmissão de vídeo volumétrico em ambientes de realidade virtual imersiva, utilizando os protocolos TCP-HTTP e WebRTC. As principais contribuições incluem desenvolvimento de ambiente emulado em hardware programável baseado nas tecnologias P4/Tofino da ferramenta P4 Programmable Patch Panel, avaliação do impacto de métricas de rede na QoE durante a transmissão de vídeo volumétrico em formato de point cloud renderizado em um dispositivo HMD real (Meta Quest 3), implementação e análise comparativa entre diferentes protocolos de transporte.*

1. Introdução

A adoção global da quinta geração de redes de comunicações móveis (5G) já abrange cerca de 2 bilhões de usuários, gerando um tráfego mensal de dados móveis estimado em 160 exabytes (EB) [Ericsson 2024]. O crescente aumento do tráfego de dados móveis em redes 5G é atribuído não somente ao uso de *smartphones*, mas também ao surgimento de diversas aplicações inovadoras que já estão pavimentando o caminho para o desenvolvimento da sexta geração de redes de comunicações móveis (6G).

Entre estas novas aplicações disruptivas destacam-se as comunicações do tipo holográfica (*Holographic-type Communications*) (HTC) [Akyildiz and Guo 2022], também comumente referenciadas por outros termos, tais como telepresença holográfica e teletransporte holográfico multi-sensorial, uma vez que podem incluir não somente os sentidos de visão e audição, mas também tato, olfato e paladar [Liberg et al. 2024]. Independentemente do termo utilizado, todos se referem ao processo completo de transmitir conteúdo holográfico para fornecer experiências imersivas em áreas tão diversas quanto jogos em nuvem, comunicações, navegação autônoma, robótica, e cidades inteligentes.

Devido à necessidade de geração, processamento e transmissão de enormes volumes de dados holográficos, as aplicações de HTC requerem tanto atraso ultra-baixo (*ultra-low delay*) quanto largura de banda ultra-alta (*ultra-high bandwidth*) [Petkova et al. 2022] para que se alcancem experiências realmente imersivas [Petkova et al. 2024]. Sendo assim, esta área de comunicações imersivas tornou-se uma das principais motivadoras para o desenvolvimento das redes 6G [Liu et al. 2023].

Devido à este grande interesse, observou-se, nos últimos anos, um aumento de estudos de qualidade de experiência (QoE) para transmissão de vídeos holográficos, também identificados como volumétricos, em aplicações de HTC. Estes estudos utilizam tanto protocolos baseados em *Transmission Control Protocol* (TCP-HTTP), tais como *Dynamic Adaptive Streaming over HTTP* (DASH), quanto em *User Datagram Protocol* (UDP), tais como *Web Real-Time Communication* (WebRTC). Dentre este grande volume de estudos, identifica-se uma lacuna a ser explorada na comparação de desempenho de QoE em ambientes de transmissão emulados incluindo a utilização pelos usuários de HMDs reais. Um estudo preliminar foi realizado em [Da Silva et al. 2024], em que se propos uma versão inicial do ambiente de testes programável para análises, porém sem comparação de desempenho de protocolos de transporte distintos. Sendo assim, as contribuições deste trabalho são:

- Desenvolvimento e disponibilização de um ambiente emulado programável, fundamentado na ferramenta P7 [Cesen et al. 2023], para transmissão de vídeo volumétrico em ambientes de realidade virtual imersiva;
- Avaliação dos impactos que métricas de rede, tais como atraso, exercem sobre a qualidade de experiência durante a transmissão de vídeos volumétricos no formato de *Point Cloud* (PC) renderizadas em um HMD real (*Meta Quest 3*);
- Implementação e comparação entre os protocolos TCP-HTTP e WebRTC.

Uma análise do estado-da-arte contendo vários estudos relacionados à transmissão de vídeos volumétricos e HTC é apresentada na seção 2. Em seguida, uma descrição do formato e da compressão de vídeo volumétrico do tipo *point cloud* é apresentada na seção 3. A seção 4 apresenta a topologia do ambiente de transmissão proposto e detalha

todos os seus componentes. Os experimentos e resultados são apresentados e discutidos na seção 5 e, finalmente, as conclusões e trabalhos futuros são apresentados na seção 6.

2. Trabalhos Relacionados

Esta seção descreve e compara diversos trabalhos recentes representando o estado-da-arte relacionado à transmissão de vídeos volumétricos. A comparação entre os trabalhos é sintetizada na Tabela 1 e abrange características referentes aos métodos de avaliação de qualidade de experiência, aos protocolos de transmissão e às técnicas de compressão (*codecs*) utilizadas.

Em [Liu et al. 2021], *Peak Signal-to-Noise Ratio* (PSNR) é utilizada como métrica objetiva de análise de QoE e não se consideram elementos fundamentais para a imersividade do ambiente virtualizado, tais como profundidade e distância.

Os autores em [Zhang et al. 2022] elaboraram um sistema projetado para melhorar transmissão de vídeo volumétrico usando super resolução 3D, integrado a mecanismos de adaptação e técnicas de aceleração. Realizaram um extensivo QoE subjetivo, com mais de 1.400 participantes remotos. Apesar de ser um framework complexo que demanda alto consumo computacional, tanto a transmissão de conteúdo volumétrico quanto a modelagem QoE subjetiva não utilizaram um HMD real, descaracterizando o cenário de realidade virtual imersiva em tempo real.

O trabalho [Udora et al. 2024] realizou uma detalhada modelagem de QoE para vídeos volumétricos capturadores por um array de três câmeras. Não foi utilizado nenhuma técnica de compressão de *point clouds*. Para aplicar as métricas disruptivas de rede foi utilizado o *netem*, que é adequado para ambientes emulados, porém a escalabilidade para experimentos reais é prejudicada.

O trabalho dos autores em [Ravuri et al. 2023] executou testes com *Partially Reliable Quick UDP Internet Connections* (QUIC) utilizando *Hypertext Transfer Protocol Third Version* (HTTP/3), o que resolve o problema de *Head of Line Block* (HOL), em uma implementação *headless* que imita a funcionalidade de um HMD. Esta abordagem, apesar de fornecer flexibilidade e simplificar o ambiente de testes, limita a validação da qualidade de experiência num ambiente de tempo real. Não foram abordados transmissão com o WebRTC ou análise comparativa de performance com o QUIC.

Os autores em [Aghaaliakbari et al. 2023] estudaram o impacto da latência e do *jitter* para transmissão de conteúdo volumétrico em um ambiente com programabilidade de *data plane* emulado. Apesar de implementarem lógica de *slices* de rede, o trabalho não fez uso de dispositivos HMD para análise de QoE imersiva em vídeos volumétricos.

Os autores em [Nguyen et al. 2023] realizaram avaliação de qualidade de transmissão de *Point Clouds* em TCP (DASH) com HMD real, porém não realizaram um estudo dos impactos de variação de métricas de rede disruptivas na QoE do usuário em ambiente imersivo, dependendo de *displays* convencionais em 2D.

O artigo em [De Fré et al. 2024] desenvolveu uma arquitetura de streaming de *point clouds* com o protocolo WebRTC aplicado na modalidade *one-to-many* na linguagem GO, usando um HMD real. A solução envolve o *Field of View* (FoV) para otimização da transmissão de vídeo volumétrico para o HMD, apresentando, assim, uma forma de auferir o QoE imersivo de maneira mais adequada. Apesar dos resultados apresentados para

QoE, não foram feitas comparações com a transmissão utilizando TCP (HTTP ou DASH).

Em [Shi et al. 2024], os autores propõe uma arquitetura de streaming projetada para otimização de qualidade de vídeo volumétrico em condições flutuantes de rede e compartimentos dinâmicos dos usuários. Também realizam a segmentação de *point clouds* em *tiles* dependentes da visão do usuário.

Tabela 1. Trabalhos relacionados à transmissão de vídeos volumétricos

Literatura	Avaliação QoE	TCP/QUIC	WebRTC	HMD Real	Compressão
Este trabalho	Subjetiva	✓	✓	✓	DRACO
[Udora et al. 2024]	Objetiva e Subjetiva	✓	✗	✓	Ausente
[De Fré et al. 2024]	Objetiva	✗	✓	✓	DRACO
[Shi et al. 2024]	Objetiva	✓	✗	✓	V-PCC
[Ravuri et al. 2023]	Objetiva	✓	✗	✗	V-PCC
[Nguyen et al. 2023]	Subjetiva	✓	✗	✓	V-PCC
[Aghaaliakbari et al. 2023]	Objetiva	✓	✗	✗	Ausente
[Zhang et al. 2022]	Objetiva e Subjetiva	✓	✗	✗	DRACO
[Liu et al. 2021]	Objetiva	✓	✗	✓	V-PCC

3. Vídeo volumétrico: Formato e Compressão de *Point Clouds*

Vídeos volumétricos, como esperados em ambientes holográficos, dependem da utilização de representações tridimensionais, tais como malhas (*meshes*), campos de luz (*light fields*) e nuvens de pontos (*point clouds - PC*), para descrever objetos dentro de um ambiente 3D. As malhas são caracterizadas por pontos interconectados por arestas para formar os polígonos que são usados para renderizar o objeto no ambiente virtual. Os campos de luz formam matrizes de imagens capturadas por câmeras plenópticas ou monoculares convencionais. Já PC consistem exclusivamente de pontos. Enquanto as malhas utilizam mapeamento de texturas, as *point clouds* usam um atributo de cor para cada ponto. Recentemente, representações de campos de radiância (*radiance fields*) também vêm sendo exploradas para renderização de ambientes 3D a partir de imagens 2D, fazendo uso de técnicas implícitas ou explícitas, tais como *3D Gaussian Splattings*.

Dentre todos os formatos disponíveis para representações tridimensionais, as *point clouds* se destacaram pela facilidade maior de captura e manipulação. Uma *point cloud voxelizada* consiste em pontos 3D mapeados em uma grade uniformemente espaçada com coordenadas de números inteiros. Cada localização de ponto (x, y, z) marca um pequeno elemento de volume cúbico chamado *voxel*. Um *voxel* pode conter um ponto (ocupado) ou estar vazio (desocupado). *Voxels* ocupados podem armazenar propriedades adicionais, tais como cor. A captura de uma *point cloud voxelizada* em um momento específico é chamado de quadro (*frame*) e o agrupamento de vários quadros juntos produz uma *point cloud voxelizada* dinâmica (vídeo em movimento).

Como as *point clouds* dinâmicas tem o potencial de fornecer seis graus de liberdade (*six degrees of freedom*) (6DoF) em movimento e uma experiência realmente imersiva, passaram a ser muito estudadas em aplicações de HTC. Devido às diferentes condições de aquisição e criação, bem como a estas aplicações finais, as características das *point clouds* podem ser muito diversas, principalmente em sua densidade. Enquanto sistemas de informação geográfica ou aplicações de navegação autônoma podem usar

point clouds bastante esparsas, aplicações de patrimônio cultural ou de realidade virtual (tais como a deste trabalho) normalmente usam *point clouds* mais densas para representar objetos e pessoas com mais precisão. Esta grande densidade requer a captura, processamento e transmissão de um número muito grande de pontos e também à obrigatoriedade de compressão deste grande volume de informação.

Os algoritmos de compressão (*codecs*) disponíveis para *point clouds* são divididos em três categorias: os baseados em projeções (tais como o *Moving Picture Experts Group* (MPEG) *Video-based Point Cloud Compression* (V-PCC), os baseados em geometria (tais como o MPEG *Geometry-based Point Cloud Compression* (G-PCC) e o Google DRACO), e mais recentemente, os baseados em redes neurais [De Fré et al. 2024]. Enquanto os *codecs* da última categoria estão em fase inicial de (crescente) evolução, os *codecs* pertencentes às outras duas categorias já vêm sendo estudados e desenvolvidos há mais de uma década, possuindo vasta literatura e padrões internacionais associados. Dentre os 3 *codecs* mencionados, o V-PCC atinge a melhor performance de compressão ao custo do maior tempo de codificação. Já o Google DRACO atinge o menor tempo de codificação, sendo possível sua utilização até mesmo para aplicações em tempo real com *point clouds* grandes, com mais de 750.000 pontos/frame, ao custo das menores taxas de compressão. Para decodificação, o Google DRACO atinge melhor desempenho para *point clouds* menores (com até 300.000 pontos/frame). Para *point clouds* maiores (tais como as densas), a performance dos *codecs* se aproxima e nenhuma atinge as restritas taxas especificadas para aplicações de HTC [Petkova et al. 2024]. Porém, é importante mencionar que as implementações utilizadas para avaliação dos *codecs* na literatura geralmente referem-se às implementações dos *softwares* de referência dos padrões de codificação, que não são otimizadas por natureza.

4. Ambiente Experimental Proposto para Transmissão de Vídeo Volumétrico

Para analisar de forma efetiva os impactos da transmissão de vídeos volumétricos para dispositivos reais em redes de comunicação, torna-se necessário um ambiente de teste que emule as condições de rede de um ecossistema real. Sendo assim, este artigo projetou e implementou um ambiente experimental baseado na topologia ilustrada na Figura 1, que representa um sistema de vídeo volumétrico como encontrado em aplicações de entretenimento e saúde, composto por um servidor, um *switch* programável, um ponto de acesso (AP) e um HMD. Esta seção detalha a seguir cada um destes componentes.

4.1. Servidor Principal

O servidor principal (*PC Server*) foi configurado com o sistema operacional Ubuntu 22.04 LTS e executou dois serviços a fim de investigar e comparar o comportamento dos protocolos TCP-HTTP e WebRTC [Alvestrand 2021] na transmissão de *point clouds* dinâmicas para dispositivos HMD reais. O WebRTC é um protocolo de comunicação UDP baseado na web, permitindo comunicação direta ponto a ponto entre navegadores e aplicativos sem a necessidade de plugins intermediários ou software proprietário.

A escolha do protocolo de transporte é chave para as aplicações de HTC. Soluções baseadas em TCP-HTTP, como DASH, não oferecem a latência em tempo real necessária para aplicações de conferências virtuais, por exemplo, devido à sobrecarga introduzida pelos mecanismos de confiabilidade de entrega, como retransmissões e reordenação de

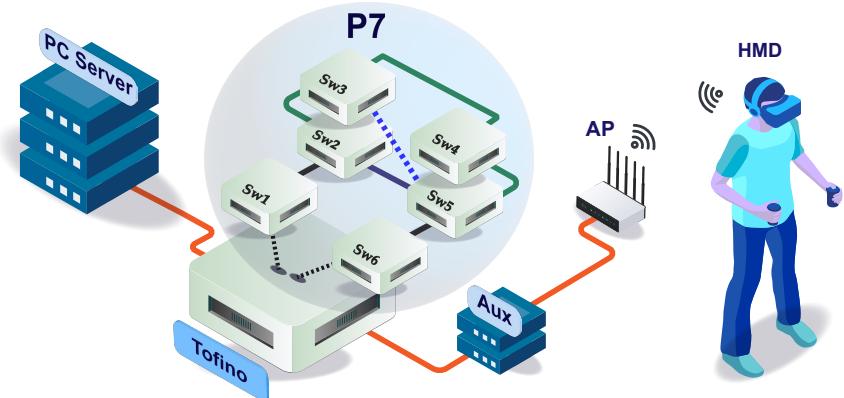


Figura 1. Visão topológica do ambiente experimental proposto para avaliação realista de QoE de vídeo volumétrico.

pacotes. Já as soluções baseadas em UDP, como WebRTC, alcançam uma latência significativamente menor, mas ao custo da falta de entregas garantidas.

Uma particularidade que diferencia os métodos que usam WebRTC e os que usam TCP-HTTP é o conceito de *Push* e *Pull* de informações. Os sistemas que usam WebRTC costumam usar *Push*, enquanto os sistemas baseados em TCP-HTTP usam *Pull*. Em sistemas baseados em *Push*, o servidor controla o fluxo de informações decidindo quais dados enviar e a que velocidade, enquanto o cliente processa as informações recebidas da melhor forma que puder. Já nos sistemas baseados em *Pull*, o cliente controla o fluxo, requisitando ao servidor os dados desejados e controlando a velocidade em que essas requisições são realizadas, enquanto o servidor tenta atendê-las da melhor forma possível.

4.2. Plataforma de Emulação P7 baseada em Switch Programável Tofino

O servidor principal (*PC Server*) é conectado ao *switch* programável Tofino através da placa de rede *Xilinx U25* de 25 Gbps. No Tofino, modelo Wedge 100BF-32X, executou-se o P7 [Cesen et al. 2023], que é uma ferramenta de emulação em hardware para experimentação de alta fidelidade em um único *switch* programável utilizando-se a linguagem *Programming Protocol-independent Packet Processors* (P4). Esta ferramenta permite ao usuário, através de um ambiente amigável semelhante ao Mininet, definir topologias, instanciar elementos de rede e características de *link* como latência, *jitter*, perda de pacotes e largura de banda. Além disso, o P7 permite a definição de um código P4 personalizado no *switch*, possibilitando ao usuário a liberdade de adicionar seu próprio *pipeline* dentro do processo de emulação da topologia. A capacidade de emulação em taxa de linha fornecida pelo P7 com programas P4 customizados a torna uma ferramenta fundamental para estudos das características de transmissão de vídeos volumétricos. A ilustração em alto nível da ferramenta pode ser observada na Figura 2.

No ambiente emulado proposto neste trabalho, o P7 é utilizado na emulação de switches programáveis e inserção de métricas disruptivas com precisão em diferentes pontos da rede. Foi executado de forma a emular seis *switches*: *Sw1*, *Sw2*, *Sw3*, *Sw4*, *Sw5* e *Sw6*, conforme ilustrado na Figura 1. Nesta composição específica, os nós *Sw1* e *Sw6* representam regiões de rede *edge* de redes 5G, ao passo que os *switches* emulados restantes representam a região *core*, interconectados de maneira a habilitar a configuração de fatias de rede (i.e., *Sw3* para o *Sw5*).

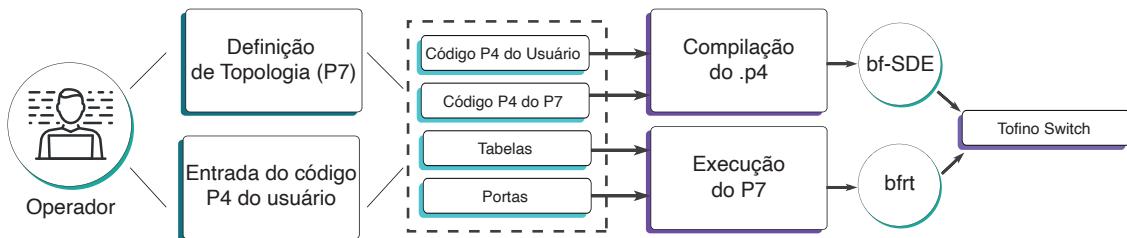


Figura 2. Arquitetura em alto nível e fluxo de trabalho do P7.

Com o objetivo de maximizar a interoperabilidade entre diferentes elementos de acesso à rede local (LAN), o *Tofino*, por meio da saída do *Sw6*, foi conectado a um servidor auxiliar (*Aux*) configurado com o sistema operacional Ubuntu 22.04 LTS e equipado com uma placa de rede *Netronome* de 25 Gbps. Esse servidor, por sua vez, foi ligado a um ponto de acesso (*AP*) Wi-Fi com uma taxa máxima de transferência de 840 Mbps. Embora esse limite represente um potencial gargalo, ele não acarretou prejuízos aos experimentos realizados, conforme apresentado na seção 5.

4.3. HMD e Renderização de *Point Clouds*

O dispositivo HMD empregado como parte do ambiente proposto foi o modelo *Meta Quest 3*, que apresenta campo de visão de 110 graus, taxa de atualização de vídeo de 120 Hz, resolução de 2064x2208 pixels por lente, chipset *Qualcomm Snapdragon XR2 Gen 2*, GPU *Adreno 740*, 8GB de RAM e sistema operacional *Horizon OS*.

Para a renderização das *point clouds* dinâmicas, utilizou-se a plataforma Unity¹ (em conjunto com a linguagem C# para scripting) para o desenvolvimento de um aplicativo para execução nativa no HMD. Foram implementadas duas versões desse aplicativo: a primeira emprega o protocolo TCP-HTTP para comunicação com o servidor, conforme descrito no Algoritmo 1, enquanto a segunda utiliza o protocolo WebRTC, conforme detalhado no Algoritmo 2. O Algoritmo 1 segue uma abordagem do tipo *Pull*, descrita anteriormente na seção 4.1, o que aumenta sua complexidade. Ele também incorpora um mecanismo de *bufferização*, permitindo carregar uma parte do vídeo enquanto outra é exibida. Por outro lado, o Algoritmo 2 é mais simples devido à implementação de vários procedimentos de controle intrínsecos ao protocolo WebRTC.

Os algoritmos mencionados foram desenvolvidos com o objetivo de permitir a comunicação eficiente entre o aplicativo e o servidor, viabilizando o recebimento de dados em formato de *point cloud* e a subsequente renderização das *point clouds* dinâmicas, que foi realizada utilizando um dos dois sistemas de partículas disponíveis no motor gráfico, o *Visual Effect Graph*. Este sistema utiliza a GPU para acelerar o processamento, permitindo um número elevado de partículas na tela (nossos testes indicaram que os óculos são capazes de suportar cerca de 200 mil partículas antes de sofrer degradação na performance). Uma solução para traduzir as listas de posições e cores de cada ponto também foi desenvolvida, onde cada partícula do sistema equivale a um ponto da nuvem. Assim, a aplicação é capaz de receber um arquivo de *point cloud* e renderizá-lo tridimensionalmente, representando fielmente as posições e cores dos pontos.

O aplicativo inclui também *plug-ins* que garantem compatibilidade com a bibli-

¹Unity Engine - <https://unity.com/products/unity-engine>

Algoritmo 1 Reprodução de vídeo via TCP

Entrada: URL

- 1: Inicializa Lista com endereços contidos na URL
 - 2: **Update:**
 - 3: **se** Buffer está vazio **então**
 - 4: Inicia co-rotina DownloadBuffer
 - 5: **fim se**
 - 6: **se** Buffer está pronto E Player está pronto **então**
 - 7: Chama função **Play**
 - 8: **fim se**
 - 9: **DownloadBuffer:**
 - 10: Envia GetRequests para cada espaço no Buffer
 - 11: Ao receber todas as respostas das GetRequests prepara Buffer
 - 12: **Play:**
 - 13: **para** int i, i menor que Tamanho do Buffer, i++ **faça**
 - 14: Exibe Point Cloud contida em Buffer[i]
 - 15: **fim para**
 - 16: Prepara player
-

Algoritmo 2 Reprodução de vídeo via WebRTC

Entrada: Conexão WebRTC

- 1: Inicializa Buffer
 - 2: **DadosRecebidosViaDataChannel:**
 - 3: **se** Dado recebido é uma string EOF **então**
 - 4: Decodifica dados do Buffer
 - 5: Coloca as posições e cores dos pontos num sistema de partículas e exibe
 - 6: **senão**
 - 7: Coloca dados recebidos no Buffer
 - 8: **fim se**
-

teca *OpenXR*, responsável pela integração com dispositivos HMD, e com a biblioteca do *codec Google Draco*, que viabiliza a leitura de arquivos no formato *.drc* diretamente na plataforma Unity.

5. Avaliação Experimental

As aplicações de HTC são sensíveis à diversas métricas de Qualidade de Serviço (QoS) em redes de comunicação. Conforme descrito em sucessivos trabalhos [Udora et al. 2024, Shi et al. 2024, De Fré et al. 2024], elas são particularmente sensíveis aos atrasos. Os trabalhos indicam que a adição de atrasos nos experimentos tem influência maior na percepção da Qualidade da Experiência (QoE) do usuário do que a introdução de perdas de pacotes, mesmo quando ocorrem re-transmissões de pacotes perdidos. De fato, até mesmo a baixa qualidade de objetos observados é mais tolerada quando o movimento destes objetos é contínuo e ininterrupto.

Não somente os atrasos relacionados às redes de transmissão, mas todas as etapas relacionadas ao *pipeline* de comunicações holográficas, tais como os *codecs*, adicionam atrasos à latência fim-a-fim ou latência de *playback*. Esta latência é definida como a diferença de tempo entre a ação que acontece na fonte e a reprodução da mesma

no usuário final. Os requisitos de latência fim-a-fim variam para diferentes aplicações de HTC. Aplicações que toleram *buffering* podem ter tolerância de latência na faixa de 50–100 ms, serviços altamente interativos podem ter como alvo 20–50 ms, e serviços de latência ultra-baixa podem ter como alvo aproximadamente 1 ms.

Dada sua relevância na QoE em aplicações de HTC, este trabalho avalia e compara, através de métricas subjetivas, o impacto de diferentes atrasos introduzidos no ambiente emulado proposto para transmissão de vídeo volumétrico. Primeiramente, esta seção apresenta as métricas de avaliação utilizadas. Em seguida, o ambiente de experimentos configurado é descrito e, então, os resultados experimentais são apresentados e discutidos.

5.1. Métricas Subjetivas de Avaliação de QoE

As avaliações de qualidade subjetiva em aplicações de transmissão de vídeo volumétrico podem ser classificadas como passivas ou interativas. No primeiro caso, os participantes do experimento fornecem suas pontuações com base em sequências de imagens que contêm visualizações fixas dos modelos 3D. No último caso, os participantes interagem de fato com os modelos 3D antes de pontuarem sua qualidade.

As avaliações de desempenho deste artigo são baseadas em um método de avaliação de qualidade interativa, uma vez que este está mais próximo do consumo natural de modelos 3D. O método de avaliação *Sequential Double Stimulus Impairment Scale (DSIS)*, descrito na recomendação ITU-R BT.500², apresenta grande acurácia para experimentos como os realizados neste artigo e foi usado na avaliação subjetiva.

5.2. Ambiente de Experimentos

De forma a se reproduzir uma aplicação imersiva de HTC para ambiente de realidade virtual, utilizou-se a *point cloud* dinâmica Long Dress [d’Eon et al. 2017]. Esta é uma *point cloud* de alta densidade e consiste de uma pessoa se movimentando sem *background*. A sequência possui 300 *frames*, gravados a 30 *frames per second* (FPS), aproximadamente 800.000 pontos/*frame* e é amplamente utilizada na literatura, com resultados de diversos experimentos disponíveis para comparação.

O ambiente de realidade virtual desenvolvido na Unity foi configurado como um estúdio de fotografia ou gravação, com o objetivo de maximizar a imersividade da experiência do usuário permitindo movimentação em 6DoF, conforme ilustram algumas capturas de tela representadas na Figura 3.

Apesar da ferramenta P7 possibilitar a definição arbitrária de métricas de QoS em todos os enlaces da rede emulada simultaneamente, neste trabalho, os valores de atraso foram inseridos somente em *Sw6* para representar uma região *edge* problemática no momento do envio dos dados para o HMD. Atrasos inseridos em outros *switches* não influenciam de forma significativa nos resultados. Para a coleta de resultados e a posterior classificação QoE, foram realizados seis testes, nos quais diferentes valores de atraso foram aplicados ao *switch* emulado *Sw6* por meio do P7: 0 ms; 1 ms; 3 ms; 5 ms; 7 ms e 10 ms. Durante essa configuração, os valores nominais utilizados refletem o atraso de uma única direção de transmissão. Desta forma, o atraso real total na topologia criada pelo P7 corresponde ao dobro do valor configurado.

²www.itu.int/rec/R-REC-BT.500



(a) Vista Frontal

(b) Vista Lateral

(c) Vista Posterior

Figura 3. Capturas de tela do HMD com ângulos da sequência Long Dress.

Seguindo a recomendação do *Sequential DSIS*, os participantes observaram primeiro a sequência transmitida via TCP-HTTP sem nenhum parâmetro de rede alterado. Depois, o vídeo foi mostrado sob diferentes condições de atraso, e os usuários avaliam as experiências correspondentes. Em seguida, foi realizada uma sequência de testes similar para o protocolo WebRTC. Aos 5 participantes foi solicitado que assistissem às sequências e avaliassem a qualidade visual utilizando uma escala MOS de 1 a 5 (1 - Péssimo, 2 - Ruim, 3 - Razoável, 4 - Bom e 5 - Excelente).

Neste ambiente de experimentos, a experiência do usuário pode ser comprometida por dois fatores distintos: a responsividade da aplicação e a fluidez da representação da *point cloud* dinâmica. O primeiro aspecto está relacionado aos processos computacionais necessários para a renderização do conteúdo no HMD, enquanto o segundo está associado principalmente à velocidade da conexão de rede. Deficiências na responsividade resultam em dessincronização entre os movimentos apresentados no dispositivo de visualização e os movimentos realizados pelo usuário, ocasionando desconforto.

5.3. Análise de Resultados

Os arquivos de captura de pacotes de rede no HMD revelaram comportamentos aparentemente paradoxais para os dois protocolos distintos que foram utilizados. O TCP-HTTP manteve uma média de *throughput* sem alterações expressivas nos seis perfis de atraso aplicados. Já o WebRTC configurado para o DataChannel acionou os mecanismos de gerenciamento de *throughput* diante de acréscimo progressivo de atraso, *GCC* (*Google Congestion Control*), que limitou o *throughput* utilizado pela transmissão do vídeo volumétrico. Isto acarretou diminuições substanciais na média de *throughput* durante a realização dos experimentos.

Os valores de *throughput* coletadas para diferentes cenários de atraso inseridos no P7 utilizando o TCP-HTTP estão ilustradas na Figura 4, ao passo que os resultados coletados com o WebRTC estão descritos na Figura 5. Além do *throughput*, as figuras fornecem leituras de média e 95% de *Confidence Interval* (CI).

No cenário sem atraso, ilustrado na Figura 4(a), o TCP-HTTP atingiu um *throughput* máximo de 112,41 Mbps, mensurado no HMD. Este desempenho é resultante das características fundamentais do protocolo, que permite o carregamento progressivo do conteúdo durante sua apresentação, mantendo a conexão estabelecida durante a transmissão de uma sequência de *frames* até a próxima requisição (REQUEST) do aplicativo

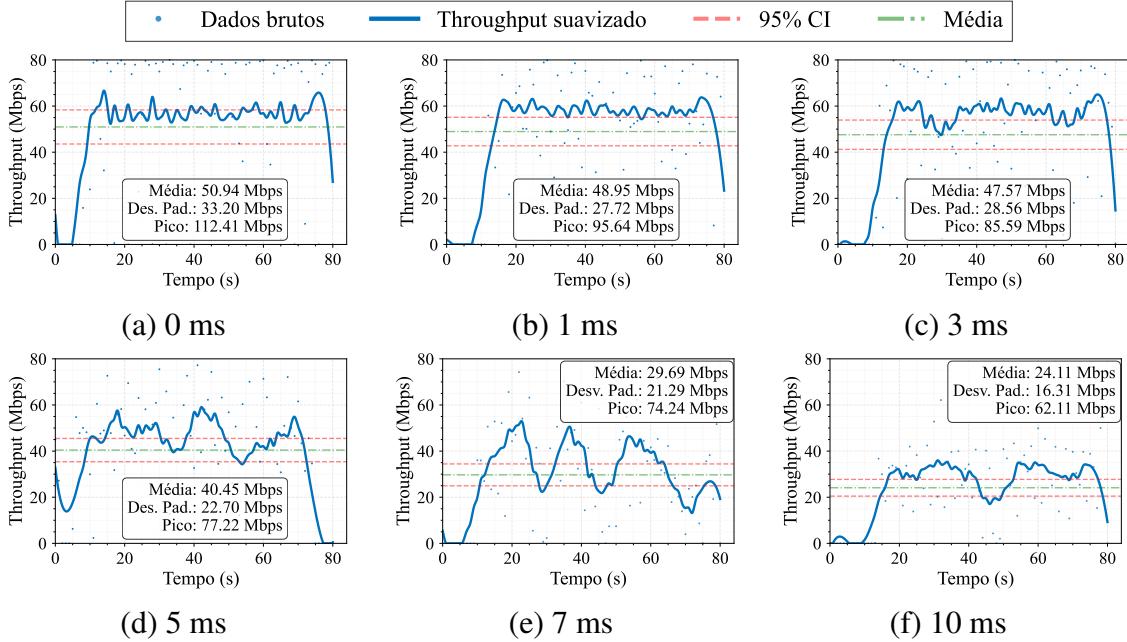


Figura 4. Comportamento do *throughput* para TCP-HTTP.

ao servidor. Os cenários com atrasos de 1 ms, 3 ms, 4 ms e 5 ms, representados nas Figuras 4(b), 4(c) e 4(d), evidenciaram uma taxa média de transmissão oscilando entre 40 Mbps e 50 Mbps. As análises dos cenários com atrasos de 7 ms e 10 ms, ilustrados nas Figuras 4(e) e 4(f), demonstraram as reduções mais expressivas no *throughput* médio em relação ao cenário ideal sem atraso medido no HMD, apresentando diminuições de 41,71% e 52,66%, respectivamente.

O WebRTC demonstrou desempenho consistentemente inferior em termos de *throughput* durante toda a execução experimental. Na Figura 5(a), observa-se o registro do valor máximo de 84,90 Mbps, com média de 43,33 Mbps. Este desempenho apresentou deterioração no cenário ilustrado na Figura 5(b), com valor médio de 22,84 Mbps, representando uma redução de 47,28% em relação ao cenário anterior.

As reduções mais expressivas entre os valores máximos e médios foram constatadas nas análises posteriores. O cenário com atraso de 3 ms, apresentado na Figura 5(c), evidenciou uma diminuição de 71,03% no *throughput* médio. Nos cenários com atrasos de 5 ms, 7 ms e 10 ms, as reduções no *throughput* médio alcançaram 80,68%, 86,66% e 87,95%, respectivamente.

A análise de desempenho é ilustrada pela Figura 6. A diferença percentual de desempenho entre os protocolos utilizados, relacionada aos valores médio de *throughput* coletados nos experimentos do ambiente de testes, está ilustrada na Figura 6(a). A linha púrpura mostra o quanto menor é o *throughput* do WebRTC em comparação ao TCP-HTTP, expresso como uma porcentagem do *throughput* do TCP-HTTP. No atraso de 10 ms, a vazão do WebRTC (5,22 Mbps) é 78,35% menor que a vazão do TCP-HTTP (24,11 Mbps), o que significa que o WebRTC opera em apenas cerca de 21,65% da capacidade do TCP-HTTP neste nível de atraso. Outros valores de diferença percentual de desempenho na Figura 6(a) são: 14,94% para 0 ms; 53,34% para 1 ms; 73,62% para 3 ms; 79,31%

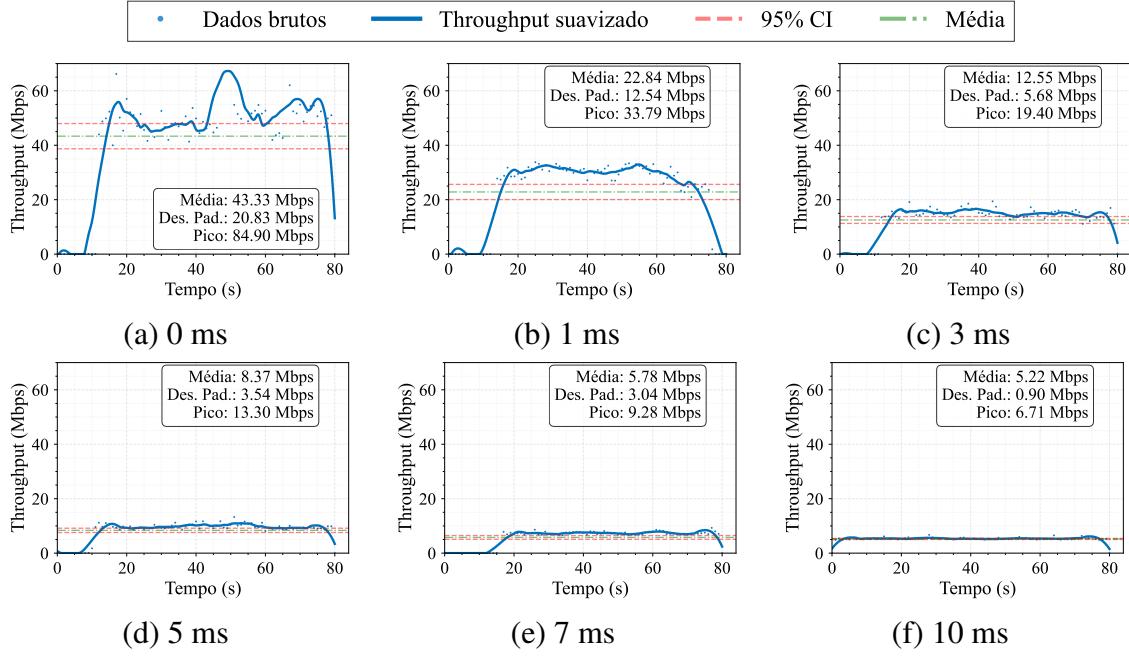


Figura 5. Comportamento do throughput para WebRTC.

para 5ms e 80,53% para 7 ms (maior que o cenário de 10 ms por apresentar amplitude de vazão superior para o TCP-HTTP).

Na Figura 6(b), estão representadas as médias das notas MOS fornecidas durante os experimentos de avaliação de qualidade subjetiva por 5 participantes do grupo de pesquisa. As avaliações máximas para o TCP-HTTP para 0 ms justificam-se pelo desempenho de throughput superior ao apresentado pela transmissão WebRTC.

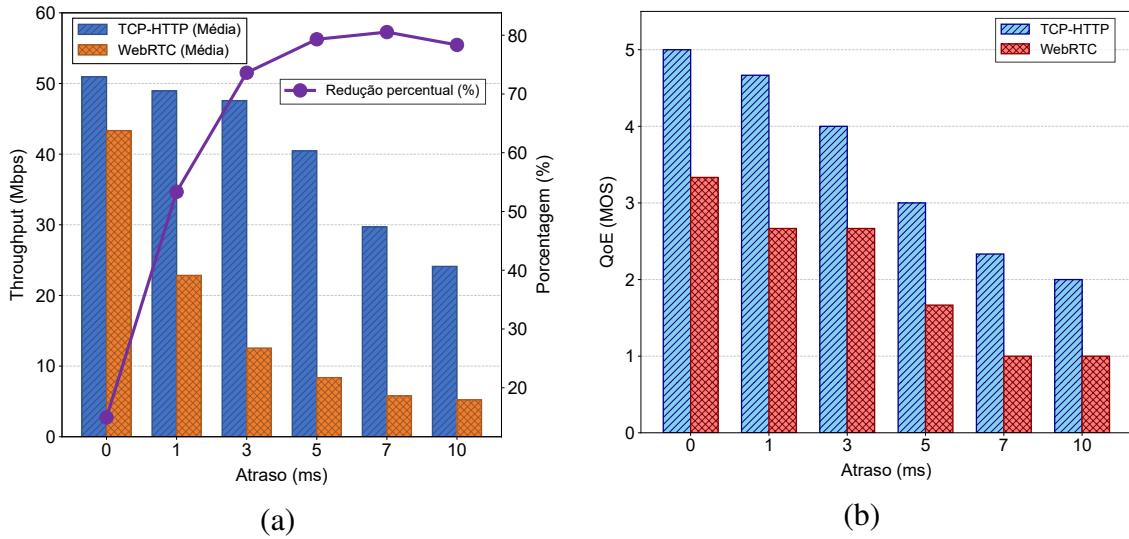


Figura 6. Análise de desempenho: (a) WebRTC em relação ao TCP-HTTP, e (b) avaliação subjetiva de QoE.

6. Conclusão e Trabalhos Futuros

Este artigo contribuiu com o desenvolvimento e com a disponibilização de um ambiente emulado programável, fundamentado na ferramenta P7, para transmissão de vídeos volumétricos em ambientes de realidade virtual imersiva. Construir um ambiente experimental não é uma tarefa trivial e de extrema importância para o avanço das pesquisas e inovações na área de HTC. Este artigo também analisa os efeitos do atraso e consequente aumento da latência na transmissão de vídeos volumétricos através dos protocolos TCP-HTTP e WebRTC, mediante a utilização de uma plataforma de emulação de redes programáveis. Os experimentos realizados evidenciaram que alterações, mesmo que sutis, na QoS da rede, são capazes de desencadear uma série de impactos negativos na QoE do usuário final, especialmente no contexto da transmissão de vídeos volumétricos do tipo *point clouds* dinâmicas.

As pesquisas futuras contemplam a configuração dinâmica de fatias de rede no P7, a qual apresenta potencial para maximizar a usabilidade deste ambiente de testes programável. Adicionalmente, propõe-se a análise comparativa de outras métricas de QoS de rede, tais como *jitter* e perda de pacotes, bem como a implementação de protocolos alternativos para comparação de desempenho, tais como o QUIC. Outro aspecto a ser investigado é a utilização de outros *codecs*, assim como o V-PCC e o G-PCC, para estabelecer comparações de desempenho com o Google Draco no mesmo ambiente de testes empregado neste trabalho.

Agradecimentos

Este trabalho foi financiado por Ericsson Telecomunicações Ltda. e pela Fundação de Pesquisa de São Paulo, bolsa 2021/00199-8, CPE SMARTNESS. Este estudo foi parcialmente financiado pela CAPES, Brazil, código de financiamento 001.

Referências

- Aghaaliakbari, F., Hmitti, Z. A., Rayani, M., Gherari, M., Glitho, R. H., Elbiaze, H., and Ajib, W. (2023). An architecture for provisioning in-network computing-enabled slices for holographic applications in next-generation networks. *IEEE Communications Magazine*, 61(3):52–58.
- Akyildiz, I. F. and Guo, H. (2022). Holographic-type communication: A new challenge for the next decade. *ITU Journal on Future and Evolving Technologies*, 3(2):421–442.
- Alvestrand, H. T. (2021). Overview: Real-Time Protocols for Browser-Based Applications. RFC 8825.
- Cesen, F. E. R. et al. (2023). Towards Multiple Pipelines Network Emulation with P7. In *IEEE NetSoft*, pages 290–292. IEEE.
- Da Silva, A. T. et al. (2024). Programmable Network Testbed for QoS/QoE Assessment of Holographic Media Delivery. In *Demo Sessions of IEEE NFV-SDN*. IEEE.
- De Fré, M., van der Hooft, J., Wauters, T., and De Turck, F. (2024). Scalable mdc-based volumetric video delivery for real-time one-to-many webrtc conferencing. In *Proceedings of the 15th ACM Multimedia Systems Conference, MMSys ’24*, page 121–131, New York, NY, USA. Association for Computing Machinery.

- d'Eon, E., Harrison, B., Myers, T., and Chou, P. A. (2017). 8i Voxelized full bodies - a Voxelized point cloud dataset. Input Document WG11M40059/WG1M74006, ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG), Geneva.
- Ericsson (2024). Ericsson mobility report. Disponível em: <https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2024>. Accessado em: 07-01-2025.
- Liberg, O., Hoymann, C., Tidestav, C., Larsson, D. C., Rahman, I., Blasco, R., Falahati, S., and Blankenship, Y. (2024). Introducing 5g advanced. *IEEE Communications Standards Magazine*, 8(1):52–57.
- Liu, R., Lin, H., Lee, H., Chaves, F., Lim, H., and Sköld, J. (2023). Beginning of the journey toward 6g: Vision and framework. *IEEE Communications Magazine*, 61(10):8–9.
- Liu, Z., Li, Q., Chen, X., Wu, C., Ishihara, S., Li, J., and Ji, Y. (2021). Point cloud video streaming: Challenges and solutions. *IEEE Network*, 35(5):202–209.
- Nguyen, M., Vats, S., van Damme, S., van der Hooft, J., Vega, M. T., Wauters, T., De Turck, F., Timmerer, C., and Hellwagner, H. (2023). Characterization of the quality of experience and immersion of point cloud videos in augmented reality through a subjective study. *IEEE Access*, 11:128898–128910.
- Petkova, R., Bozhilov, I., Manolova, A., Tonchev, K., and Poulkov, V. (2024). On the way to holographic-type communications: Perspectives and enabling technologies. *IEEE Access*, 12:59236–59259.
- Petkova, R., Poulkov, V., Manolova, A., and Tonchev, K. (2022). Challenges in implementing low-latency holographic-type communication systems. *Sensors*, 22(24).
- Ravuri, H. K., Vega, M. T., Van Hooft, J. D., Wauters, T., and De Turck, F. (2023). Adaptive partially reliable delivery of immersive media over quic-http/3. *IEEE Access*, 11:38094–38111.
- Shi, Y., Clement, B., and Ooi, W. T. (2024). Qv4: Qoe-based viewpoint-aware v-pcc-encoded volumetric video streaming. In *Proceedings of the 15th ACM Multimedia Systems Conference, MMSys '24*, page 144–154, New York, NY, USA. Association for Computing Machinery.
- Udora, C., Qian, P., Anmulwar, S., Fernando, A., and Wang, N. (2024). Quality of experience modelling and analysis for live holographic teleportation. In *2024 International Conference on Computing, Networking and Communications (ICNC)*, pages 598–604.
- Zhang, A., Wang, C., Han, B., and Qian, F. (2022). YuZu: Neural-Enhanced volumetric video streaming. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 137–154, Renton, WA. USENIX Association.