UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA – PPGCAP

EDENILSON JÔNATAS DOS PASSOS

BALANCEAMENTO DE CARGA PARA OTIMIZAR A TRANSMISSÃO DE VÍDEOS MPEG-DASH

JOINVILLE
2024

EDENILSON JÔNATAS DOS PASSOS

BALANCEAMENTO DE CARGA PARA OTIMIZAR A TRANSMISSÃO DE VÍDEOS MPEG-DASH

Dissertação apresentada ao Programa de Pós–Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Adriano Fiorese

JOINVILLE
2024

dos Passos, Edenilson Jônatas
Balanceamento de carga para otimizar a transmissão de vídeos MPEG-DASH / Edenilson Jônatas dos Passos. - Joinville, 2024.
76 p. : il. ; 30 cm.

Orientador: Adriano Fiorese.
.
Dissertação (Mestrado) - Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada, Joinville, 2024.

1. Balanceamento de Carga. 2. SDN. 3. MPEG-DASH. I. Fiorese, Adriano. II. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada. III. Solução de Balanceamento de Carga para Otimizar a Distribuição de Conteúdo em Transmissão de Vídeo MPEG-DASH.

EDENILSON JÔNATAS DOS PASSOS

BALANCEAMENTO DE CARGA PARA OTIMIZAR A TRANSMISSÃO DE VÍDEOS MPEG-DASH

Dissertação apresentada ao Programa de Pós–Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Adriano Fiorese

BANCA EXAMINADORA:

Dr. Adriano Fiorese
UDESC

Membros:

Dr. Roger Kreutz Immich
UFRN

Dr. Adriano Fiorese
UDESC

Dr. Guilherme Piêgas Koslovski
UDESC

Joinville, 08 de Julho de 2024

AGRADECIMENTOS

Agradeço ao meu orientador por aceitar conduzir o meu trabalho de pesquisa. A todos os meus professores do curso de da Universidade do Estado de Santa Catarina – Udesc pela excelência da qualidade técnica de cada um.

Aos meus pais que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória. Sou grato à minha família pelo apoio que sempre me deram durante toda a minha vida.

Deixo um agradecimento especial ao meu orientador pelo incentivo e pela dedicação do seu escasso tempo ao meu projeto de pesquisa.

RESUMO

Nos últimos anos, o paradigma de negócios relacionado aos serviços de transmissão de conteúdo multimídia sob demanda através da Internet tem demonstrado grande potencial. Vários fatores contribuíram para a popularidade desse modelo, sendo a capacidade de acessar mídia preferida a qualquer momento e em qualquer lugar um dos aspectos mais destacados. No entanto, garantir uma experiência de usuário satisfatória tem sido um desafio considerável para os provedores de serviços devido às demandas exigentes dessas transmissões de conteúdo, especialmente aquelas de natureza audiovisual. Mesmo com a alocação significativa de recursos computacionais, a preocupação com sobrecarga continua a ser um problema recorrente. Uma abordagem que se mostra promissora para abordar essa questão é o uso de balanceadores de carga, que têm o objetivo de equilibrar de forma uniforme a carga de trabalho entre as unidades de processamento. Este estudo propõe uma solução para esse desafio, utilizando uma combinação de redes definidas por software (SDN) para otimizar a utilização de recursos de processamento e transmissão de dados e melhorar a experiência do usuário final. A abordagem envolve o monitoramento contínuo, pelo controlador de rede, das métricas de desempenho, como utilização de CPU, RAM, disco e throughput dos servidores de conteúdo. Com base nesses dados, o controlador decide qual servidor está mais apto a receber uma nova conexão. Caso essa verificação ocorra durante a reprodução de um vídeo, a abordagem permite a migração da conexão entre servidores de forma transparente, sem que o usuário perceba a mudança. Os resultados demonstram que a solução proposta proporciona melhorias no tempo de resposta, alcançando até 80% mais agilidade e consistência na qualidade da transmissão de vídeo. A análise também mostrou que, ao distribuir eficientemente a carga de trabalho entre os diversos servidores e ajustar dinamicamente a transmissão de conteúdo, a abordagem reduz a latência e evita interrupções durante a reprodução de vídeo. Além disso, a técnica de redirecionamento de conexões não impactou negativamente a qualidade percebida da transmissão, garantindo uma experiência de usuário fluida e de alta qualidade. Esta pesquisa não só confirma a viabilidade da aplicação de SDN em serviços de transmissão de vídeo sob demanda, mas também estabelece uma base para futuras melhorias e adaptações em ambientes de rede.

Palavras-chave: Balanceamento de Carga, SDN, MPEG-DASH.

ABSTRACT

In recent years, the business paradigm related to on-demand multimedia content streaming services over the Internet has shown great potential. Several factors have contributed to the popularity of this model, with the ability to access preferred media at any time and from anywhere being one of the most highlighted aspects. However, ensuring a satisfactory user experience has been a considerable challenge for service providers due to the demanding nature of these content streams, especially those of an audiovisual nature. Despite significant allocation of computational resources, concerns about overload remain a recurring issue. A promising approach to addressing this issue is the use of load balancers, which aim to evenly distribute the workload among processing units. This study proposes a solution to this challenge by using a combination of software-defined networks (SDN) to optimize the use of processing and data transmission resources and improve the end-user experience. The approach involves continuous monitoring by the network controller of performance metrics such as CPU usage, RAM, disk, and throughput of content servers. Based on the collected data, the controller determines which server is best suited to handle a new connection. If this assessment occurs during video playback, the approach allows for seamless migration of the connection between servers without the user noticing the change. The results demonstrate that the proposed solution provides improvements in response time, achieving up to 80% greater speed and consistency in video transmission quality. The analysis also showed that by efficiently distributing the workload among various servers and dynamically adjusting content transmission, the approach reduces latency and prevents interruptions during video playback. Additionally, the connection redirection technique did not negatively impact the perceived quality of the transmission, ensuring a smooth and high-quality user experience. This research not only confirms the feasibility of applying SDN in on-demand video streaming services but also establishes a foundation for future improvements and adaptations in network environments.

Keywords: Load Balancing. SDN. MPEG-DASH.

LISTA DE ILUSTRAÇÕES

LISTA DE ACRÔNIMOS

ABR	*Adaptive Bitrate Streaming*
ACK	*Acknowledgement*
CDN	*Content Distribution Networks*
CPU	*Central Process Unit*
DASH	*Dynamic Adaptive Streaming over HTTP*
DNS	*Domain Name Server*
DWRR	*Dynamic Weighted Round Robin*
FIN	*Finalize*
HEVC	*High Efficiency Video Coding*
HLS	*HTTP Live Streaming*
HTTP	*Hypertext Transfer Protocol*
IEC	*International Electrotechnical Commission*
IoT	*Internet of Things*
ISO	*International Organization for Standardization*
Kbps	*Kilobits Per Second*
LFU	*Least Frequently Used*
LRU	*Least Recently Used*
LTS	*Long Term Support*
Mbps	*Megabits Per Second*
MPD	*Multimedia Presentation Description*
MPEG	*Moving Picture Expert Group*
NBI	*Northbound Interface*
ONF	*Open Networking Foundation*
QoE	*Quality of Experience*
QoS	*Quality of Service*
RAM	*Random Access Memory*
RST	*Reset*
RTT	*Round-Trip Time*
SBI	*Southbound Interface*
SDN	*Software Defined Networking*

SYN	*Synchronous*
TCP	*Transmission Control Protocol*
URL	*Uniform Resource Locator*
VP9	*Video Predictor 9*
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Os mercados de computação em nuvem e transmissão de vídeo por assinatura estão em constante crescimento nos últimos anos. Com o advento da pandemia de COVID-19, esses números se tornaram ainda mais significativos. Segundo a *Global Industry Analysts* (ANALYSTS, 2021) serviços de computação em nuvem tiveram uma receita de aproximadamente US$313.1 bilhões em 2020 e ainda, estimam que até o ano de 2027 esse número pode alcançar cerca de US$947 bilhões.

Sendo assim, a área de serviços de transmissão de vídeo apresenta características semelhantes quanto ao crescimento dos serviços de nuvem computacional, pois segundo (STOLL, 2021) no ano de 2020 esse segmento de mercado conquistou aproximadamente US$100 bilhões e sugere que até 2026 esse número supere US$200 bilhões. Além disso, segundo (SANDVINE, 2023), o conteúdo relacionado a vídeos representou 65,93% do volume total pela Internet.

Haja visto tamanho crescimento, o espaço para novas tecnologias e abordagens quanto a infraestrutura de rede de transmissão de dados é promissor, uma vez que provedores de serviço apresentam dificuldade para lidar com a crescente demanda e manter a qualidade de serviço (*Quality of Service* (QoS)) adequada. Nesse contexto, há métodos para atenuar tal problema. Um deles é o balanceamento de carga dos serviços. Segundo (HARIS; KHAN, 2022), a escolha ótima de provedor (i.e., computador, servidor ou de serviço) feita pelo balanceador pode beneficiar o sistema como um todo em diversos fatores como por exemplo, na redução da chance de falha e sobrecarga proporcionando robustez, melhora da escalabilidade, redução do tempo de resposta de maneira geral e dessa forma, resultando na maior satisfação do cliente e reduzindo consideravelmente o custo de manutenção do sistema.

O balanceamento de carga pode ser endereçado por diversas abordagens. Uma delas é a baseada em métricas de desempenho. De mogo geral, diante de uma requisição, o modelo de balanceamento de carga seleciona o servidor mais apto para atender a demanda naquele momento. Esse processo de escolha do servidor pode ser baseado em uma ou mais métricas pré-definidas ou no comportamento da rede como um todo.

Neste trabalho, é apresentada uma abordagem para a distribuição equitativa de carga de processamento, por meio do redirecionamento de tráfego de requisição e resposta de conteúdo de vídeo, centrada na monitorização e subsequente recuperação das métricas constituintes do indicador de balanceamento de carga através da adoção do paradigma de Redes Definidas por Software (*Software Defined Networking* (SDN)), viabilizando um método de balanceamento de carga pautado em métricas, operando diretamente na camada de sessão da infraestrutura de rede. Isto é, ao invés de depender de um servidor específico para o balanceamento, a alocação equitativa da carga de atendimento dos clientes é executada nos próprios dispositivos de comutação de pacotes compatíveis com a arquitetura OpenFlow (SDN), situados na porção de rede onde se localizam os diversos servidores organizados em *cluster*. Para tanto, é realizado redirecionamento do tráfego de requisições dos clientes para os servidores de conteúdo adequados à política de

balanceamento de carga.

1.1 DEFINIÇÃO DO PROBLEMA

De acordo com (OMER; MOHAMMEDEL-AMIN; MUSTAFA, 2021), uma das estratégias para lidar com a alta demanda e distribuir a carga de trabalho entre os servidores em uma rede de distribuição de conteúdo é a aplicação de técnicas que envolvem a atualização periódica das configurações de *Domain Name Server* (DNS), ajustando os endereços correspondentes às requisições. No entanto, a utilização dessas abordagens baseadas em DNS apresenta desafios consideráveis quando se trata de conexões de longa duração e fluxos contínuos, como é o caso do tráfego de conteúdos multimídia. Nestes cenários, uma vez que um servidor é designado, o conteúdo é, geralmente, transmitido a partir de um único servidor através de requisições sequenciais que abrangem fluxos contínuos e exigem alta taxa de transferência de dados. Como consequência, esse padrão operacional dificulta a capacidade de resposta a eventos como picos repentinos na demanda e também contribui para a ocorrência de falhas operacionais.

Neste contexto, Wang (WANG; HUANG; ROSE, 2018) destaca diversas questões relacionadas às estratégias de sistemas de balanceamento de carga baseados em DNS, pois as *Content Distribution Networks* (CDN) que adotam essa abordagem enfrentam desafios recorrentes. Estes desafios emergem da falta de consideração adequada da geolocalização do cliente, o que pode resultar na alocação de um servidor que não se mostra ideal para atender aquele cliente naquele momento específico. Como resultado, a qualidade da experiência do usuário pode ser prejudicada.

Dessa forma, este trabalho propõe uma abordagem para o problema do balanceamento de carga em servidores de conteúdo multimídia, com foco em conexões com alto tráfego de dados e longa duração, visando criar um método mais eficaz e responsivo às demandas operacionais.

1.2 OBJETIVOS

Nesta seção serão apresentados os objetivos geral e específicos deste trabalho.

1.2.1 Objetivo Geral

O objetivo geral do presente trabalho é criar um sistema abrangente de otimização da entrega de conteúdo de vídeo *Moving Picture Expert Group* (MPEG)-*Dynamic Adaptive Streaming over HTTP* (DASH), unindo o projeto, desenvolvimento e implementação de um mecanismo eficiente de balanceamento de carga de atendimento de clientes.

Isso será alcançado por meio da criação de uma estrutura hierárquica de servidores de conteúdo idênticos, integrando nós na borda para o armazenamento de cache. Esta infraestrutura operará em um ambiente SDN e tirará proveito dos recursos oferecidos por uma CDN. A meta

principal é a gestão dinâmica das características dos servidores, assegurando a seleção apropriada para otimizar a transmissão do conteúdo de vídeo de maneira eficaz.

1.2.2 Objetivos Específicos

Os seguintes objetivos específicos estão elencados na elaboração deste trabalho:

- Realizar uma análise da literatura pertinente assim como comparar os trabalhos relacionados a conceitos de balanceamento de carga em redes de distribuição de conteúdo multimídia;
- Realizar uma análise da literatura para identificar as métricas de desempenho mais relevantes e impactantes no contexto de redes de distribuição de conteúdo;
- Elaborar e implementar um sistema de redirecionamento de conexões *Transmission Control Protocol* (TCP) dinâmico por meio da aplicação de tecnologias de redes definidas por software;
- Elaborar o cenário de testes com auxílio do ambiente de experimentação real distribuído Cloudlab;
- Coletar, interpretar e apresentar os dados obtidos através dos testes.

1.3 METODOLOGIA

Este estudo empreende uma investigação de aplicação prática, fundamentada primordialmente na adoção de métodos de pesquisa referenciada, juntamente com a análise dos resultados obtidos. No contexto da escala de maturidade proposta por (WAZLAWICK, 2020) os tipos de pesquisa em Computação podem ser classificados em cinco níveis. O nível 1 se caracteriza pela apresentação de um produto, onde se descreve um novo produto ou solução já o nível 2 é caracterizado pela apresentação de algo diferente, destacando novidades em relação ao estado da arte. O nível 3 traz a apresentação de algo presumivelmente melhor, sugerindo melhorias teóricas sem provas empíricas robustas. O nível 4 é descrito como apresentação de algo reconhecidamente melhor, com evidências concretas e empíricas de superioridade e por último, o nível 5 é caracterizado pela apresentação de uma prova, oferecendo validação formal ou matemática da nova abordagem. Este estudo se posiciona no terceiro nível, sugerindo avanços superiores, mas limitado pela ausência de testes padronizados que permitiriam comparações mais abrangentes entre as abordagens documentadas na literatura.

Quanto à categorização do raciocínio lógico, conforme delineado por (LAKATOS, 2021), este estudo se encaixa no paradigma hipotético-dedutivo, uma vez que emprega testes não padronizados como meio de validar sua abordagem.

1.4 ORGANIZAÇÃO DO TEXTO

A estrutura deste trabalho é delineada da seguinte maneira: O Capítulo 2 introduz o embasamento teórico, elucidando os conceitos fundamentais necessários para a compreensão da pesquisa em questão, incluindo tópicos como redes definidas por software (SDN), redes de entrega de conteúdo (CDN), balanceamento de carga e distribuição de conteúdo multimídia. Ademais, esse capítulo apresenta um panorama dos estudos correlatos à temática em foco. No Capítulo 3, detalha-se a arquitetura proposta para a avaliação da abordagem proposta. No Capítulo 4, são expostos os experimentos conduzidos e a análise dos resultados obtidos a partir da abordagem proposta. Por fim, o Capítulo 5 aborda as conclusões extraídas e as possíveis direções para pesquisas futuras, ressaltando o potencial de aprofundamento nas áreas exploradas.

2 EMBASAMENTO TEÓRICO

Neste capítulo, são delineados os fundamentos essenciais para a apreensão da abordagem proposta. Inicialmente, aborda-se os conceitos pertinentes às redes definidas por software, seguidos pela análise das redes de distribuição de conteúdo, com destaque para o padrão MPEG-DASH. Em seguida, discutem-se os aspectos relacionados aos métodos e métricas de balanceamento de carga, culminando na consideração crítica do papel do cache em ambientes de sistemas distribuídos. Por fim, os trabalhos relacionados são apresentados.

2.1 REDES DE COMPUTADORES

Redes de computadores (TANENBAUM; WETHERALL, 2011) são sistemas essenciais que permitem a interconexão e comunicação entre dispositivos. Compostas por nós, protocolos, topologias e camadas organizacionais, as redes de computadores viabilizam o compartilhamento de informações e recursos de maneira eficiente e confiável. Além disso, a segurança da informação desempenha um papel crítico na proteção dos dados e da integridade da rede, enquanto a administração e o gerenciamento são necessários para manter o funcionamento adequado. A Internet, uma rede global de redes, exemplifica o impacto transformador das redes de computadores, permitindo o acesso à informação e serviços em escala global. Portanto, compreender os princípios e componentes das redes de computadores é essencial para acompanhar a evolução tecnológica e a interconectividade que caracterizam a era digital.

Em uma rede tradicional, o tráfego de dados é gerenciado através de uma infraestrutura fixa e hierárquica de dispositivos de rede, como roteadores e *switches*. Esses dispositivos utilizam configurações e regras para determinar como os dados devem ser encaminhados, fazendo parte do que é conhecido como plano de controle. Esse plano de controle é responsável por estabelecer as diretrizes e políticas que governam o fluxo de informações na rede.

Além disso, paralelamente ao plano de controle, surge o conceito de plano de dados. O plano de dados é a camada operacional que efetivamente encaminha os pacotes de dados seguindo as instruções previamente definidas pelo plano de controle. Ele é responsável por executar as ações práticas de roteamento e comutação dos dados, garantindo que eles alcancem seu destino final com eficiência e segurança.

2.1.1 Redes Definidas por Software

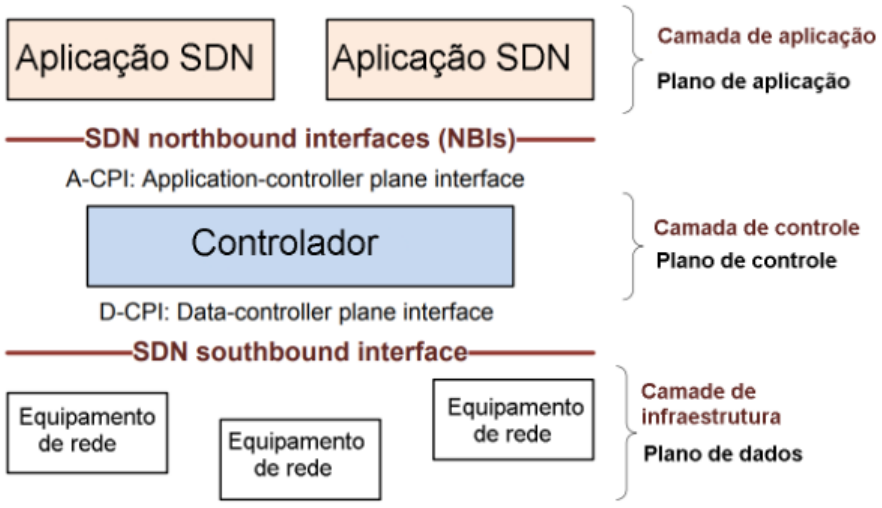
A arquitetura de Redes Definidas por Software se diferencia das redes tradicionais devido à separação entre o plano de controle e o plano de dados dos dispositivos de rede. Essa separação oferece diversas vantagens, destacando-se a centralização do controle sobre os dispositivos de rede (COMER, 2016). Para viabilizar o funcionamento efetivo de uma rede SDN, é essencial a implementação de um protocolo de comunicação entre os dispositivos envolvidos. O protocolo OpenFlow, que teve origem em 2008 sob a autoria de McKeown (McKeown et al., 2008), tornou-

se um padrão consolidado em 2011, quando a *Open Networking Foundation* (ONF) (ONF, 2015) assumiu a responsabilidade pela sua padronização. Mais de 90 empresas colaboraram com a ONF para aprimorar o desenvolvimento desse protocolo. Este avanço foi crucial para a evolução e adoção das redes SDN.

A arquitetura de Rede Definida por Software (SDN), conforme apontado por (ONF, 2014), busca proporcionar uma interface aberta que permita o desenvolvimento de software capaz de controlar conexões de rede, bem como o fluxo de tráfego que atravessa essas conexões, incluindo a capacidade de inspecioná-lo e modificá-lo. Um dos principais componentes de uma rede SDN é o controlador que é o cérebro da rede, responsável por tomar decisões de roteamento e encaminhamento com base em políticas definidas pelo administrador da rede (GORANSSON; BLACK, 2014). Também é responsável por traduzir as políticas em regras específicas para os dispositivos de rede. Outro componente importante é o plano de dados que é composto pelos dispositivos de rede, como *switches* e roteadores, que encaminham o tráfego de acordo com as instruções do controlador SDN.

A interface que permite que aplicativos e serviços de gerenciamento interajam com o controlador SDN é conhecida como *Northbound Interface* (NBI). Ela fornece uma maneira para os administradores de rede definirem políticas, monitorarem o desempenho e automatizarem tarefas de gerenciamento. A mesma situação ocorre com a interface entre o controlador SDN e os dispositivos de rede no plano de dados que é conhecido como *Southbound Interface* (SBI). Ela permite que o controlador envie instruções de encaminhamento e colete informações sobre o estado da rede dos dispositivos de rede. As instruções necessitam de protocolos para que o controlador SDN e os dispositivos de rede se comuniquem, como o OpenFlow, que é um dos protocolos mais amplamente utilizados em ambientes SDN. A Figura 1 ilustra como a arquitetura funciona.

Figura 1 – Arquitetura SDN



Fonte: *Open Networking Foundation, 2014*

2.1.2 OpenFlow

O protocolo OpenFlow, um dos pilares fundamentais da arquitetura SDN, teve sua origem em 2008 e foi formalmente estabelecido em 2011 pela *Open Networking Foundation (ONF)*. A ONF assumiu a responsabilidade de padronizar o protocolo, contando com o apoio de mais de 90 empresas, incluindo gigantes da tecnologia como Facebook, Google, Microsoft e Verizon, para promover seu desenvolvimento e adoção (PERRIN; HUBBARD, 2013).

De acordo com (McKeown et al., 2008), o protocolo OpenFlow é baseado no funcionamento de um comutador Ethernet, com uma tabela de fluxo interna e uma interface padronizada para adicionar e remover entradas de fluxo. Na arquitetura SDN, ele é utilizado na interação entre o controlador e o plano de dados, funcionando como uma Interface de Controle-Planejamento de Dados (*D-CPI*).

Os elementos de rede, como os *switches*, efetuam o encaminhamento de pacotes utilizando uma ou mais tabelas de fluxo. Essas tabelas contêm regras de fluxo utilizadas para verificar os cabeçalhos dos pacotes, determinar as ações a serem efetuadas e manter contadores de ações. A atuação do controlador em relação a essas regras pode ser tanto proativa, instalando regras conforme os pacotes chegam, quanto reativa, caso haja uma notificação de evento, como um pacote que não teve correspondência na tabela de fluxos (KLöTI; KOTRONIS; SMITH, 2013). Assim, o OpenFlow exemplifica a flexibilidade e a capacidade de gerenciamento dinâmico que a arquitetura SDN oferece, permitindo uma administração mais eficiente e adaptável das redes modernas.

2.2 TRANSMISSÃO ADAPTATIVA DE TAXA DE BITS

A transmissão adaptativa de taxa de bits (SANI; MAUTHE; EDWARDS, 2017), do inglês *Adaptive Bitrate Streaming* (ABR), é uma técnica avançada utilizada na entrega de conteúdo multimídia. Essa abordagem permite que o conteúdo, seja ele vídeo ou áudio, se ajuste dinamicamente de acordo com a largura de banda disponível ao dispositivo utilizado pelo usuário final.

O funcionamento da ABR envolve a pré-codificação do conteúdo em várias qualidades, ou seja, em diferentes taxas de bits. Esse conteúdo é então dividido em segmentos chamados de *chunks* (segmentos), geralmente com duração de 1 a 8 segundos cada. Quando o usuário inicia a reprodução do conteúdo desejado, o reprodutor monitora continuamente a qualidade da conexão de Internet. Se houver uma diminuição na largura de banda disponível, o reprodutor automaticamente ajusta a requisição dos próximos *chunks* para a qualidade adequada em tempo real, garantindo uma reprodução suave e sem interrupções. Da mesma forma, se a conexão do usuário melhorar, a ABR oferece a possibilidade de transmitir o conteúdo em uma qualidade superior, proporcionando uma experiência de visualização ou audição mais rica e envolvente.

Existem diversos formatos de ABR amplamente utilizados, sendo os mais comuns o MPEG-DASH e o *HTTP Live Streaming* (HLS) (Bitmovin, 2024). Esses formatos permitem uma

implementação eficaz da transmissão adaptativa de taxa de bits em uma variedade de plataformas e dispositivos, tornando-a uma técnica essencial para a entrega de conteúdo multimídia de alta qualidade pela Internet (SANI; MAUTHE; EDWARDS, 2017).

2.2.1 MPEG-DASH

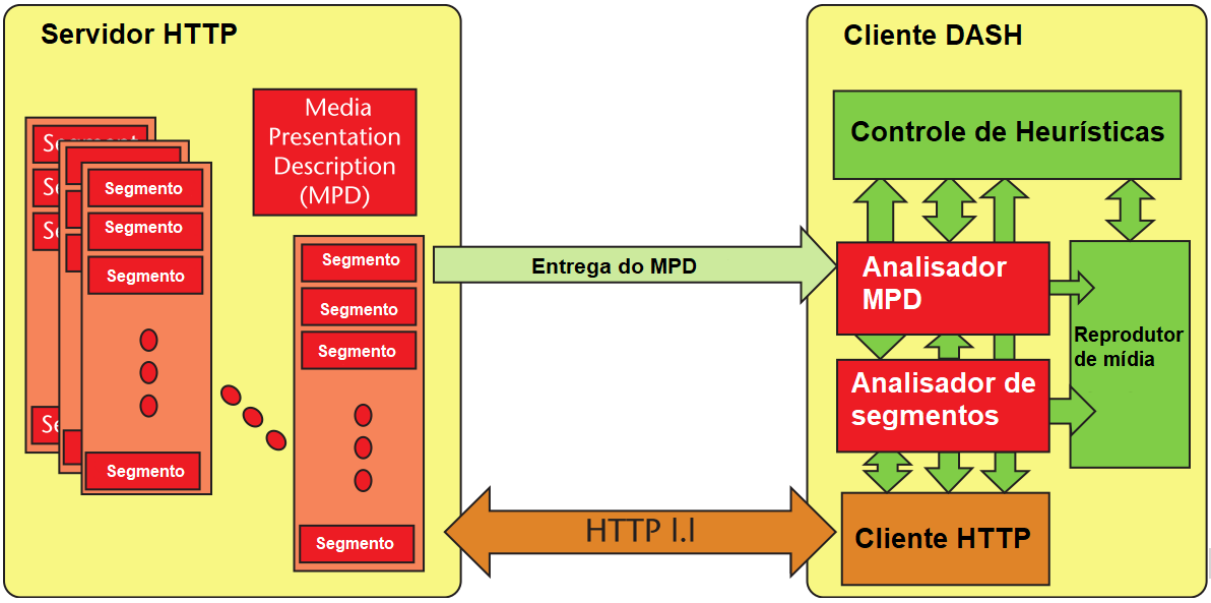
O padrão MPEG-DASH, padronizado pela *International Organization for Standardization* (ISO)/*International Electrotechnical Commission* (IEC) 23009-1 (ISO, 2014), representa uma abordagem inovadora para a transmissão de vídeo pela Internet, utilizando o protocolo *Hypertext Transfer Protocol* (HTTP) como base. Desde sua introdução em 2011, este padrão tem demonstrado um desempenho altamente eficaz no que diz respeito à capacidade de adaptação durante a transmissão de conteúdo multimídia.

Essencialmente, o MPEG-DASH opera por meio da segmentação do conteúdo de vídeo em unidades de arquivo individualmente codificadas, que são hospedadas em um servidor web. Para facilitar a seleção apropriada desses segmentos, com base nas condições da rede do espectador, é disponibilizado um arquivo de manifesto, geralmente em formato *Extensible Markup Language* (XML) ou *Multimedia Presentation Description* (MPD), que contém informações cruciais sobre as diferentes taxas de bits e resoluções disponíveis para o vídeo em questão. É fundamental observar que o MPEG-DASH é agnóstico em relação ao *codec*, o que implica dizer que ele é capaz de interoperar com uma variedade de *codecs* de vídeo e áudio, proporcionando, assim, flexibilidade significativa na entrega de conteúdo multimídia. O mérito de sua adaptabilidade e ampla compatibilidade resultou em sua crescente adoção na indústria, tornando-o um elemento central no cenário da transmissão de conteúdo digital.

A Figura 2 apresenta um esquema da estrutura do padrão MPEG-DASH que inicia sua operação com o cliente DASH procurando, inicialmente, o arquivo de descrição de mídia (MPD) no servidor de conteúdo especificado. A partir deste arquivo, o cliente realiza uma análise para extrair informações cruciais sobre as propriedades do conteúdo disponível. Essas informações englobam o tipo de conteúdo oferecido, as diferentes resoluções (qualidades) disponíveis, os requisitos de largura de banda mínima e máxima para cada taxa de bits disponível e, por fim, os endereços dos segmentos do arquivo. Usando essa riqueza de dados, o cliente DASH é capaz de determinar a melhor correspondência entre suas próprias capacidades de largura de banda e as características de qualidade e tipo de conteúdo disponíveis.

É fundamental ressaltar que o cliente DASH desempenha um papel crucial no processo pois é responsável por tomar decisões informadas sobre a qualidade e os segmentos a serem solicitados. Isso é alcançado através da análise de sua largura de banda atual em relação às especificações do conteúdo, permitindo que escolha a qualidade e inicie as solicitações de segmentos por meio do método GET do protocolo HTTP. Assim, esse aspecto do padrão MPEG-DASH se destaca como um diferencial significativo, pois coloca a responsabilidade pelo gerenciamento de qualidade e seleção de arquivos nas mãos do cliente DASH, proporcionando uma experiência de *streaming* adaptativa otimizada.

Figura 2 – Estrutura MPEG-DASH



Fonte: Adaptado de SODAGAR, 2011

2.2.2 Codecs

Um codec, ou codificador-decodificador, é uma ferramenta fundamental no universo da compressão de vídeo. Essa tecnologia desempenha um papel crucial na eficiente transmissão de conteúdo multimídia pela Internet. Devido à alocação significativa da largura de banda disponível na rede para o consumo de vídeo, os codecs de compressão de vídeo surgem como uma alternativa viável para mitigar os desafios relacionados à congestão de rede. Isso se torna ainda mais evidente devido aos esforços contínuos para melhorar a qualidade de vídeo, como a adoção de resoluções mais altas, como os vídeos 4K e 8K, que requerem taxas de bits elevadas e, consequentemente, uma largura de banda substancial para transmissão. A compressão de vídeo, portanto, se propõe a reduzir o tamanho dos arquivos multimídia originais sem comprometer, ou minimamente afetar, sua qualidade final. Atualmente, há alguns codecs de compressão de vídeo que mais que se destacam. Entre eles estão o *High Efficiency Video Coding* (HEVC) e o *Video Predictor 9* (VP9).

O codec HEVC é capaz de comprimir o tamanho de um arquivo de vídeo em até 50%, proporcionando uma significativa economia de largura de banda. Um nível semelhante de compactação é alcançado pelo codec VP9. Diante das considerações sobre economia de largura de banda e a entrega de conteúdo de alta qualidade aos usuários, muitos projetos e iniciativas têm adotado a tecnologia de compressão. Isso não apenas reduz a carga sobre as redes de transmissão, mas também garante uma experiência de visualização de vídeo mais eficiente e satisfatória para os usuários. Essa abordagem desempenha um papel fundamental na otimização do uso de recursos de rede, contribuindo para uma distribuição de vídeo mais eficaz na era da Internet de alta velocidade.

2.3 BALANCEAMENTO DE CARGA

O balanceamento de carga é um sistema que tem como principal objetivo distribuir de forma equitativa a carga de trabalho entre os dispositivos envolvidos. Nesta seção são apresentadas as classificações de balanceamento de carga com atuações estáticas e dinâmicas, incluindo os principais algoritmos utilizados nas soluções de balanceamento de carga mais populares nos dias atuais.

2.3.1 Balanceamento de Carga Estático

A aplicação do balanceamento de carga estático envolve a distribuição de carga de maneira fixa, sem levar em consideração as mudanças nas condições do ambiente (HAMADAH, 2017). Isso significa que as atribuições de tarefas ou recursos são definidas antecipadamente e não se ajustam automaticamente com base na demanda em constante evolução.

Uma das principais características do balanceamento de carga estático é sua previsibilidade. Em ambientes onde a carga de trabalho é estável e bem conhecida, como em redes locais de pequeno porte ou em sistemas de computação com aplicativos de demanda relativamente constante, o balanceamento de carga estático pode ser uma abordagem eficaz. Isso ocorre porque as condições previsíveis permitem que os administradores de sistemas aloquem recursos de forma a otimizar o desempenho geral do sistema.

No entanto, o balanceamento de carga estático apresenta limitações significativas em ambientes mais dinâmicos e heterogêneos. Em cenários nos quais a carga de trabalho varia ao longo do tempo, surgem desafios adicionais. Por exemplo, em ambientes de nuvem, sites de comércio eletrônico, ou redes sociais, a demanda pode variar consideravelmente com base em fatores como horário do dia, eventos sazonais, ou lançamento de produtos. Nesses casos, um sistema de balanceamento de carga estático não consegue otimizar a alocação de recursos de forma eficiente, resultando em subutilização de recursos em momentos de baixa demanda e possíveis sobrecargas em picos de atividade. Um dos algoritmos de balanceamento de carga estático mais populares é o Round Robin.

2.3.1.1 Round Robin

O Round Robin é um algoritmo de balanceamento de carga simples e amplamente utilizado que distribui solicitações ou tarefas de maneira equitativa entre os servidores disponíveis. Funciona de forma cíclica, em que cada servidor recebe uma solicitação por vez na ordem em que estão listados (KILLELEA, 2002). Esse método é eficaz quando os servidores têm recursos semelhantes ou similares e a carga de trabalho é uniformemente distribuída. No entanto, o Round Robin não leva em consideração a capacidade de processamento de cada servidor, o que pode resultar em desempenho desigual em ambientes onde os servidores têm diferentes características de processamento.

Uma variação do Round Robin que lida com essa questão é o *Weighted Round Robin* (Round Robin Ponderado). Nesse método, a cada servidor é atribuído um peso ou prioridade com base em sua capacidade de processamento. Os servidores com maior capacidade recebem pesos mais altos, o que significa que eles receberão uma parcela maior da carga de trabalho (WANG; CASALE, 2014). Isso ajuda a evitar sobrecargas em servidores mais fracos e garante uma distribuição mais ajustada, levando em consideração a capacidade de processamento individual de cada servidor.

2.3.2 Balanceamento de Carga Dinâmico

Em contraste com o balanceamento de carga estático, o balanceamento de carga dinâmico é uma abordagem que se adapta continuamente às mudanças nas condições do ambiente e às variações na carga de trabalho (AFZAL; GANESH, 2019). Esta estratégia é mais utilizada em ambientes onde as demandas dos usuários ou as características da carga de trabalho são altamente variáveis e imprevisíveis.

Uma das principais motivações para a implementação do balanceamento de carga dinâmico é a necessidade de otimizar a utilização dos recursos de hardware e software disponíveis, a fim de garantir um desempenho consistente e eficiente em ambientes de computação distribuída. Por exemplo, um serviço de transmissão de vídeos online experimenta picos de tráfego durante eventos ao vivo, como eventos esportivos ou transmissões de *shows*. Nesse contexto, um algoritmo de balanceamento de carga dinâmico pode identificar os servidores menos carregados em tempo real e direcionar as solicitações dos usuários para esses servidores, evitando sobrecargas e garantindo uma experiência de usuário fluida.

Além disso, o balanceamento de carga dinâmico desempenha um papel crucial em ambientes de computação em nuvem, onde a alocação eficiente de recursos é essencial para otimizar custos e garantir a escalabilidade. Um exemplo onde o balanceamento de carga dinâmico á adequado é em uma empresa que hospeda seus aplicativos na nuvem e que experimenta flutuações significativas na demanda ao longo do dia. Nesse cenário, algoritmos de balanceamento de carga dinâmico podem alocar automaticamente mais recursos de computação durante os períodos de pico e liberar recursos durante os períodos de menor demanda, economizando recursos e dinheiro.

2.3.2.1 Round Robin Ponderado Dinamicamente

O algoritmo Round Robin Ponderado Dinamicamente *Dynamic Weighted Round Robin* (DWRR) é uma estratégia de balanceamento de carga dinâmica levando em consideração a carga atual de cada nó. A principal inovação do DWRR em relação ao Round Robin convencional é a atribuição dinâmica de pesos aos recursos com base em sua utilização. Isso significa que, por exemplo, a medida que os recursos de computação se tornam mais ocupados, seus pesos são ajustados para refletir essa carga adicional. Assim, os servidores menos sobrecarregados

recebem mais solicitações em relação aos mais congestionados, resultando em uma distribuição mais eficiente da carga de processamento. O DWRR é uma abordagem versátil e adaptável que contribui significativamente para melhorar o desempenho e a confiabilidade de sistemas de rede em ambientes dinâmicos e com variação na carga de trabalho.

2.3.3 Outras Estratégias de Balanceamento Dinâmico

Estes algoritmos desempenham um papel crucial na otimização do desempenho das redes de distribuição de conteúdo. Eles podem ser classificados em três categorias distintas, cada uma com suas próprias características e critérios de seleção.

A primeira categoria engloba os algoritmos que utilizam a distância de saltos na rede como principal critério de seleção de um servidor. Essa abordagem busca minimizar a latência e melhorar a eficiência da entrega de conteúdo, selecionando servidores com menor número de saltos na topologia da rede.

A segunda categoria de algoritmos se baseia em métricas de desempenho. É importante destacar que a escolha do servidor desempenha um papel crítico na infraestrutura de uma rede de distribuição de conteúdo baseada em replicação. Nesse contexto, os algoritmos levam em consideração métricas previamente definidas ou o comportamento geral da rede para determinar qual servidor será utilizado. Ferramentas especializadas são empregadas para coletar e analisar essas métricas, contribuindo para a tomada de decisões assertivas.

Um exemplo significativo de métrica utilizada nesse contexto é a latência do cliente em relação ao servidor. Quanto menor for a latência, maior a probabilidade de que o servidor selecionado entregue o conteúdo de forma rápida e eficaz ao cliente. Portanto, a minimização da latência é um objetivo chave ao escolher o servidor apropriado para atender às demandas dos usuários.

Além dessas duas categorias principais, a terceira categoria de algoritmos de seleção de servidores também desempenha um papel importante, empregando abordagens mais complexas, como aprendizado de máquina e análise preditiva, para otimizar a escolha do servidor com base em critérios dinâmicos e em tempo real.

2.3.3.1 Métricas de Desempenho

Avaliar o desempenho de um servidor, especialmente de um servidor de conteúdo multimídia, é de extrema importância, pois permite monitorar e determinar seus limites de utilização. Existem diversas abordagens para analisar o desempenho de um servidor. Seguindo a perspectiva de (WICHTLHUBER; REINECKE; HAUSHEER, 2015), as três principais métricas incluem:

- **Taxa de Conexões por Segundo** - Mede quantas conexões o servidor pode atender em um intervalo de tempo específico, geralmente em segundos.

- **Número de Conexões Simultâneas** - Refere-se ao total de clientes conectados ao servidor ao mesmo tempo. Isso também pode ser expresso como a capacidade do servidor em lidar com conexões TCP simultâneas.

- ***Throughput*** - Indica a máxima taxa de transmissão que o servidor pode oferecer. Seguindo a definição fornecida pela Oracle (Oracle, 2007), a taxa de transferência pode ser compreendida como a medida da carga de trabalho que um servidor é capaz de suportar, sendo especificamente expressa pelo número de requisições processadas por minuto por instância do servidor.

Acerca do *throughput* é possível elencar algumas particularidades, enquanto métrica para o balanceamento de carga. À medida que o número de usuários que acessam o servidor aumenta, é observado um crescimento correspondente no *throughput*. Esse fenômeno é considerado um padrão característico dessa métrica. Portanto, em uma perspectiva inicial, a busca por um servidor com um alto *throughput* é desejável, uma vez que isso implica na capacidade de atender a um maior volume de requisições por minuto. No entanto, é crucial reconhecer que existe um limite para a escalabilidade do servidor. A partir de um determinado ponto, o acréscimo de mais clientes conectados resultará em uma deterioração dessa métrica.

Este comportamento revela um dilema inerente à otimização do *throughput*. Embora a busca por uma alta taxa inicial seja compreensível, é imperativo considerar os limites de capacidade do servidor. Superar esses limites pode levar a um decréscimo na eficiência do *throughput*, impactando negativamente o desempenho global do sistema. Portanto, a análise da escalabilidade do servidor e a compreensão de seus limites tornam-se fatores críticos ao buscar otimizar o *throughput* em ambientes de alta concorrência.

Além, das 3 métricas previamente citadas conforme (WICHTLHUBER; REINECKE; HAUSHEER, 2015), outras também se prestam a aferir o desempenho da rede ou dos servidores, e dessa forma auxiliar no balanceamento de carga. Entre elas pode-se elencar:

Latência: Em redes de computadores, a latência representa o tempo que um pacote leva para percorrer o caminho entre um dispositivo e seu destino final. Em algumas referências, ela é também denominada latência ponto a ponto, e pode ser interpretada como o atraso na chegada de um pacote (delay) (Oracle, 2003).

Perda de Pacotes: Representa a porcentagem de pacotes de dados perdidos no trânsito entre o cliente e o servidor. A alta perda de pacotes pode causar interrupções e artefatos defeituosos no vídeo.

Número de Conexões Ativas: Trata-se do número de conexões ativas em cada servidor. Um servidor com muitas conexões pode ficar sobrecarregado levando à degradação do desempenho.

CPU e Memória RAM: O monitoramento da utilização da *Central Process Unit* (CPU) e memória *Random Access Memory* (RAM) também é importante pois a alta utilização desses recursos pode indicar um servidor com alta carga de trabalho, o que pode afetar sua capacidade de lidar com novas conexões de maneira eficiente.

E/S para dispositivo de armazenamento: A quantidade de dados que estão sendo lidos e gravados nas unidades de armazenamento de dados do servidor. Alta E/S de de armazenamento persistente de dados pode levar à degradação do desempenho e a gargalos.

2.3.3.2 Limitações das Métricas de Desempenho

As métricas de desempenho tem um papel fundamental no monitoramento do estado de um servidor, especialmente em contextos críticos, como servidores de conteúdo multimídia. No entanto, é crucial reconhecer que a análise isolada de cada métrica pode não proporcionar uma avaliação completa e precisa do desempenho do sistema. Um exemplo ilustrativo disso é o *throughput*, que, embora seja uma métrica relevante, por si só, não é suficiente para determinar o desempenho global de um servidor de conteúdo multimídia.

O *throughput*, que mede a quantidade de dados transferidos entre o servidor e os clientes em um determinado período, é apenas um dos muitos fatores a serem considerados. Ignorar outras métricas críticas, como latência e perda de pacotes, pode levar a uma avaliação incompleta e até enganosa do desempenho do servidor. No contexto de conteúdo multimídia, a latência e a perda de pacotes são particularmente importantes, pois têm um impacto direto na experiência do usuário.

Portanto, é evidente que a abordagem mais eficaz para avaliar o desempenho de servidores de conteúdo multimídia envolve a análise conjunta de diversas métricas de desempenho. Isso permite uma compreensão mais abrangente e precisa do estado do servidor e possibilita a identificação e correção eficaz de problemas que podem afetar a qualidade da entrega de conteúdo multimídia. Além disso, a utilização de métricas complementares, como latência e perda de pacotes, é essencial para garantir uma experiência de usuário consistente e de alta qualidade em aplicações de transmissão de vídeo e áudio.

2.3.4 Balanceamento de Carga e Balanceamento de Tráfego

Uma distinção fundamental se faz necessária entre os termos "balanceamento de carga"e "balanceamento de tráfego", uma vez que, embora possam parecer semelhantes, se referem a conceitos distintos. O balanceamento de carga é um conceito mais amplo, que se refere à distribuição de cargas de trabalho entre vários recursos, enquanto o balanceamento de tráfego é uma forma específica de balanceamento de carga que se concentra na distribuição de tráfego de rede (BOURKE, 2001).

O balanceamento de carga pode ser usado para distribuir qualquer tipo de carga de trabalho, incluindo processamento de dados, armazenamento de dados e outras formas de

computação. O balanceamento de tráfego, por outro lado, é usado especificamente para distribuir tráfego de rede.

Em termos práticos, o balanceamento de carga é geralmente implementado por meio de um dispositivo de hardware ou software chamado balanceador de carga. O balanceador de carga monitora o estado dos recursos disponíveis e distribui o tráfego de acordo com esses recursos. Enquanto o balanceamento de carga se concentra mais na distribuição de solicitações entre servidores, o balanceamento de tráfego está relacionado à distribuição de dados ao longo de rotas de rede.

Essa distinção é crucial para a compreensão e implementação de estratégias eficazes de otimização e escalabilidade de sistemas, especialmente em ambientes de rede complexos e com alto volume de tráfego. Portanto, ao analisar esses conceitos, é essencial reconhecer suas diferenças e entender como cada abordagem contribui para a eficiência e o desempenho de infraestruturas de rede e servidores.

2.4 COMPUTAÇÃO DE BORDA

A computação de borda ou na borda, também conhecida como *edge computing*, é uma abordagem que se concentra em levar o processamento de dados e a capacidade de computação para mais perto das fontes de dados, como dispositivos *Internet of Things* (IoT), sensores e outros dispositivos de rede (SHI et al., 2016). Em contraste com a computação em nuvem tradicional, que centraliza o processamento em *data centers* remotos, a computação de borda distribui tarefas de processamento de forma mais eficiente, reduzindo a latência e melhorando a capacidade de resposta em tempo real.

No contexto de distribuição de conteúdo isso é fundamental para aplicações de transmissão de vídeo, onde a latência e a capacidade de resposta são fatores determinantes para a qualidade da experiência do usuário (YOSHIHISA, 2021). Ao distribuir tarefas de processamento de vídeo e armazenamento em cache nos nós mais próximos do usuário, essa tecnologia permite uma entrega mais rápida e eficiente de conteúdo multimídia.

Além disso, a computação de borda pode reduzir significativamente a carga nos servidores centrais e na infraestrutura de rede principal, distribuindo o tráfego de dados de forma mais equilibrada. Isso resulta em uma melhor utilização dos recursos e uma redução no risco de sobrecarga nos pontos centrais da rede. Com essa abordagem, é possível oferecer uma experiência de transmissão mais suave e confiável, mesmo durante picos de demanda.

Contudo, implementar computação de borda no sistema de distribuição de conteúdo envolve a integração de diversas tecnologias e práticas, incluindo o uso de dispositivos para coleta de dados em tempo real, algoritmos de otimização para direcionar o tráfego de dados de forma eficiente, e mecanismos de cacheamento para armazenar conteúdos populares nos nós de borda (BACCOUR et al., 2019). Essa combinação de tecnologias permite que as redes de distribuição de conteúdo melhorem a qualidade do serviço e aumentem a satisfação dos usuários

finais.

2.5 CACHE EM SISTEMAS DISTRIBUÍDOS

O uso de cache desempenha um papel fundamental na otimização do desempenho de sistemas distribuídos e na melhoria da experiência do usuário ao acessar informações na Internet. Cache refere-se à prática de armazenar cópias de dados frequentemente acessados em locais mais próximos do usuário a fim de reduzir a latência e acelerar o acesso a esses dados (MONTEIRO et al., 2020). Esta técnica é essencial para enfrentar o desafio da latência, especialmente quando os computadores e servidores estão geograficamente dispersos.

Uma das estratégias mais comuns para implementar o cache é a criação de servidores de cache. Estes servidores mantêm cópias dos dados mais solicitados, agindo como intermediários entre o cliente e o servidor principal. Ao armazenar dados em servidores de cache, é possível reduzir a carga no servidor principal, acelerando as respostas às solicitações do cliente.

Diferentemente das réplicas, que são geralmente planejadas com antecedência e mantêm uma cópia completa do conjunto de dados, o cache é uma abordagem sob demanda. Isso significa que os dados são armazenados ou atualizados no cache à medida que os usuários solicitam informações específicas. Essa abordagem dinâmica torna possível priorizar os dados mais relevantes e frequentemente acessados, economizando recursos e melhorando a eficiência.

Em uma abordagem empírica amplamente adotada, inúmeras infraestruturas de distribuição de conteúdo são construídas a partir de um arranjo complexo de caches em múltiplas camadas, frequentemente dispostos em uma estrutura hierárquica que evoca analogia à arquitetura em árvore (KARAMCHANDANI et al., 2016). Dentro desse contexto, o servidor de origem assume a posição de nó raiz, enquanto os utilizadores finais se encontram associados aos caches mais periféricos. Essa configuração de múltiplas camadas proporciona uma variedade de benefícios, que vão desde otimizações de latência até a redução do tráfego direto ao servidor de origem, promovendo assim uma eficiente distribuição de conteúdo.

Um exemplo notável do uso de cache é encontrado no serviço de *streaming* Netflix®. A Netflix® é uma aplicação distribuída que utiliza extensivamente a técnica de cache para melhorar a qualidade do serviço e fornecer *streaming* de vídeo de alta qualidade pela Internet. Ao armazenar em cache os vídeos, metadados e informações de perfil dos usuários em servidores localizados em várias regiões geográficas, a Netflix® consegue reduzir a latência e garantir que os usuários tenham acesso rápido aos conteúdos desejados.

2.5.1 Políticas de Gestão de Cache

Uma consideração importante na implementação de políticas de cache é a gestão do cache, que envolve decisões sobre o que armazenar em cache, por quanto tempo e como atualizar ou invalidar o cache quando os dados subjacentes mudam. Políticas de cache eficazes são essenciais

para garantir que o cache seja útil e não gere dados desatualizados ou desnecessários. A seguir são listadas as políticas de cache mais comuns.

2.5.1.1 Cache Baseado em Localização

A elaboração de estratégias para determinar o conteúdo a ser armazenado em um cache regional envolve uma série de desafios complexos. De acordo com (DERNBACH et al., 2016) esse processo requer a utilização de métricas sofisticadas que avaliem de maneira precisa a afinidade das preferências dos usuários dentro de uma determinada localidade, ao mesmo tempo em que identificam as discrepâncias entre diferentes regiões geográficas. Além disso, um modelo de preferência do usuário deve levar em consideração a natureza híbrida das preferências dos usuários em uma região específica, que tipicamente envolvem uma combinação de conteúdo amplamente popular em escala global e conteúdo específico da região. Essa abordagem sugere que políticas de cache híbrido que levem em conta tanto o conteúdo globalmente popular quanto o conteúdo regionalmente popular podem oferecer benefícios significativos no contexto do armazenamento de dados em caches regionais.

Adicionalmente, é importante salientar que a seleção de conteúdo para um cache regional não é apenas uma questão de popularidade, mas também deve levar em conta a relevância para os usuários daquela região específica. Portanto, a análise das preferências dos usuários e o ajuste das políticas de cache devem ser constantes, para garantir uma experiência de usuário otimizada.

2.5.1.2 Least Recently Used

O método de Substituição do Menos Recentemente Usado (*Least Recently Used* (LRU)) é um algoritmo de substituição de cache que remove o item menos recentemente acessado do cache quando um novo item precisa ser adicionado. A ideia fundamental por trás do LRU é que se um determinado item não foi utilizado por um longo período, é menos provável que seja utilizado no futuro próximo (LUO; CHANGSHENG; CHENGFENG, 2007). Portanto, ele se torna um candidato ideal para substituição quando o cache está cheio.

O funcionamento em etapas acontece da seguinte maneira:

1. Quando um novo item precisa ser adicionado ao cache, o algoritmo de gerenciamento de cache verifica se o cache está cheio.
2. Se o cache não estiver cheio, o novo item é simplesmente adicionado ao cache.
3. Se o cache estiver cheio, o algoritmo identifica o item menos recentemente usado. Isso é o item que não foi acessado por mais tempo.
4. O item menos recentemente usado é então removido do cache para liberar espaço para o novo item.
5. O novo item é adicionado ao cache.

O método LRU é utilizado em sistemas de cache para otimizar o uso da memória (OLANREWAJU et al., 2016), garantindo que os itens mais relevantes sejam mantidos no cache, enquanto os menos usados são removidos para acomodar novos dados. Este algoritmo é eficaz e eficiente na maioria das situações de gerenciamento de cache. Esta política é simples, mas nem sempre é a mais eficaz para *streaming* de vídeo, pois não considera a popularidade ou a importância do conteúdo.

2.5.1.3 Least Frequently Used

A estratégia de gestão de cache Usado com Menos Frequência (*Least Frequently Used* (LFU)) tem como objetivo principal monitorar a frequência de acesso a cada item e, quando o cache atinge sua capacidade máxima, substituir o item menos frequentemente acessado (HASSLINGER et al., 2018). Esta política de cache é notavelmente centrada na popularidade do conteúdo, tornando-se particularmente relevante em cenários de sistemas de *streaming* de vídeo, onde se observa uma discrepância na popularidade de diferentes vídeos disponíveis. Essa abordagem se destaca por sua capacidade de otimizar o desempenho do cache, garantindo que os recursos mais demandados permaneçam disponíveis, melhorando assim a experiência do usuário em serviços de *streaming* de vídeo.

É importante ressaltar a diferença entre o LFU e o LRU. Uma página que foi menos recentemente usada também pode ser a menos frequentemente usada. No entanto, isso não é uma regra geral, e os algoritmos focam em critérios diferentes para tomar decisões de substituição.

2.5.1.4 Pre-fetching

O *pre-fetching* (tradução literal pré-carregamento) é uma estratégia adotada que antecipa quais conteúdos os utilizadores possivelmente solicitarão a seguir, com base em seu histórico ou padrões de visualização (CUCCHIARA; PICCARDI; PRATI, 2004), por exemplo. Essa técnica consiste em armazenar o conteúdo em um cache antecipadamente, visando minimizar as interrupções durante a reprodução. Pode ser implementada em diversos estágios do processo de *streaming* de vídeo. A abordagem de *pre-fetching* do lado do cliente ocorre no dispositivo do usuário, onde os segmentos de vídeo são proativamente armazenados em cache pelo navegador. Por outro lado, o *pre-fetching* no servidor ocorre no próprio servidor, onde os segmentos de vídeo são enviados ao dispositivo do usuário antecipadamente, antes mesmo de serem solicitados. Esta abordagem tem como objetivo otimizar a experiência do usuário, garantindo uma transição mais suave e eficiente entre os conteúdos solicitados. Além disso, o *pre-fetching* contribui para a redução do tempo de espera, melhorando, assim, a eficiência do sistema em questão.

2.6 TRABALHOS RELACIONADOS

A identificação de trabalhos relacionados foi feita por meio de motores de busca (Google, IEEEXplore, ACM) com os seguintes termos, não necessariamente nesta ordem. Porém, todas

as palavras de cada item devem aparecer nos resultados (artigos), seja no título, resumo ou no corpo:

- *SDN Video Load Balancing*;
- *Video Load Balancing Using SDN*;
- *SDN DASH Load Balancing*;
- *SDN Load Balancing VOD*.

Os trabalhos são excluídos caso possuam assuntos aliados às temáticas listadas a seguir. Ao excluir tais palavras-chave é possível reduzir o número de resultados irrelevantes que podem surgir e, assim, economizar tempo na triagem e análise de trabalhos garantindo uma seleção mais precisa e eficiente dos trabalhos relacionados.

- *5G*;
- *6G*;
- *VANETs*;
- *IoT*;
- *Wi-Fi*.

O resultado dessa pesquisa foi filtrado para incluir apenas os trabalhos que apresentam semelhanças com a solução proposta. Ao restringir o escopo aos trabalhos mais pertinentes, se busca identificar padrões, destacar avanços tecnológicos e metodológicos aplicáveis, e compreender melhor as limitações e oportunidades de melhoria na área de estudo.

Assim, o trabalho de (LIU; WANG; ZHANG, 2020) trata de otimizar a distribuição de conteúdo de vídeo em redes de computação na borda. Para isso, propõe uma arquitetura de rede de controle centralizada chamada *Named Data Networking* (NDN). Nesta abordagem, nós são implantados nas bordas da rede para fornecer armazenamento em cache e capacidades de processamento, melhorando a qualidade de experiência e economizando largura de banda. Ainda, naquele é discorrido sobre a distribuição eficiente de conteúdo de vídeo, a segmentação de vídeos em diferentes categorias e o uso de uma rede definida por software (SDN) para gerenciamento e controle centralizados.

O artigo de (MAJDABADI; WANG; RAKAI, 2022) propõe um *framework* de otimização baseado em DASH para aprimorar a *Quality of Experience* (QoE) no *streaming* DASH. O *framework* de otimização maximiza o número de sessões de *streaming* simultâneas que podem ser acomodadas em uma rede e a qualidade do *streaming*. A implementação prática do *framework* utiliza o roteamento dinâmico e alocação de largura de banda possibilitados pela rede definida por software.

O *Content Steering* é uma técnica proposta por (GROUP, 2022) que visa otimizar a entrega de conteúdo de vídeo pois possibilita rotas dinâmicas entre diferentes redes de distribuição de conteúdo. Essa técnica é viabilizada por meio do *Multimedia Presentation Description* (MPD), um componente chave do padrão *Dynamic Adaptive Streaming over HTTP* (DASH), que descreve a estrutura do conteúdo de vídeo e os recursos disponíveis para sua entrega. O MPD pode incluir múltiplos endereços *Uniform Resource Locator* (URL) para o mesmo segmento de vídeo, cada um apontando para servidores em diferentes CDNs. Um orquestrador central, que monitora a qualidade da conexão e o desempenho das diferentes CDNs, pode, com base nessas informações, tomar decisões em tempo real sobre qual CDN utilizar a qualquer momento.

A pesquisa de (ANDJAMBA; ZODI, 2023) introduz o Protocolo de Roteamento de Balanceamento de Carga (LBRP) como um protocolo de roteamento adaptativo que melhora a experiência do usuário em aplicações de streaming de vídeo. O LBRP realiza o roteamento considerando parâmetros como capacidade de enlace e re-roteamento de tráfego para enlaces subutilizados. O desempenho do LBRP foi testado usando Mininet, levando em consideração métricas como distribuição de largura de banda, throughput e latência. Os resultados mostram que o LBRP dá precedência ao tráfego em tempo real (*livestream*) sobre o tráfego sob demanda, proporcionando uma melhor experiência de visualização para o usuário .

A pesquisa de (TAHA, 2023) propõe um algoritmo baseado em inteligência artificial para alterar as direções dos pacotes em redes SDN. O modelo proposto estima o custo dos caminhos dados nas redes com base em cinco critérios: tamanho do pacote de rede, números de pacotes, intervalo de tempo total necessário, capacidade do enlace (largura de banda) e número de saltos (caminho mais curto). Dessa forma, os caminhos ótimos do remetente para o destinatário podem ser facilmente determinados. Esse mecanismo permite ao controlador SDN minimizar o tempo de decisão necessário para selecionar os fluxos. Com base nos critérios mencionados, foi criado um conjunto de dados que contém informações sobre atraso de roteamento. Do modelo proposto, três critérios - tamanho, número e tempo do pacote - foram utilizados para encontrar o atraso ótimo do pacote a ser utilizado posteriormente no modelo para encontrar o custo de cada caminho. Uma comparação de referência entre o estado da arte e o algoritmo sugerido revela que o tempo de consumo para selecionar um caminho de recuperação ótimo apresenta uma redução significativa no atraso, estimada em alguns milissegundos. Consequentemente, pode reduzir os caminhos de gargalo e a utilização de recursos.

Outro trabalho que se assemelha, (BAMHDI, 2023) propõe uma arquitetura híbrida que combina Redes de Comunicação Centradas em Informações (ICN) e Redes Definidas por Software (SDN) para criar um sistema de cache transparente na rede para distribuição de conteúdo sobre a rede IP tradicional. A arquitetura visa melhorar o desempenho de serviços de vídeo sob demanda (VoD) para clientes, enquanto utiliza eficientemente os recursos do provedor de rede. Um protótipo chamado CDCA foi desenvolvido e avaliado em um ambiente de emulação Mininet. Os resultados da avaliação demonstram que a arquitetura híbrida CDCA para criar um sistema de cache para distribuição de conteúdo aprimora o desempenho do serviço VoD e

otimiza a utilização de recursos de rede.

Em (BUKHARI; AFAQ; SONG, 2023) é apresentado um método para prever recursos em um *switch* em uma rede baseada em SDN. Para isso, um cenário de transmissão de vídeo é implantado em uma rede SDN e métricas de desempenho são registradas. Os recursos são previstos usando quatro algoritmos de aprendizado de máquina. Mais especificamente, o artigo propõe uma implementação de teste de um cenário de transmissão de vídeo para avaliar o desempenho da abordagem proposta.

O estudo apresentado por (CHIANG; LI, 2024) propõe uma arquitetura de serviço de Cache Estendido para Redes Definidas por Software (ESC - *Extended SDN Cache*). O ESC desagrega a função de inspecionar o tráfego de entrada, tomar decisões de cache e armazenar conteúdo em três entidades de rede diferentes, reduzindo assim a carga de uma única entidade de rede. Além disso, para reduzir a carga do controlador SDN, é utilizado um switch OpenFlow estendido chamado switch de DPI (*Deep Packet Inspection*), que pode inspecionar o tráfego de entrada. O ESC projetou um mecanismo que pode armazenar em cache diferentes partes de um vídeo em nós de cache distintos, aumentando assim a capacidade de cache e a flexibilidade do sistema. Os resultados da análise mostram que o atraso médio de espera do ESC é menor do que os outros dois métodos comparados, C-flow e OpenCache.

Analisando as pesquisas mais recentes que abordam o balanceamento de carga em sistemas de distribuição de conteúdo, com foco especial no contexto da transmissão de vídeo, é evidente que existem oportunidades significativas para aprimoramentos. Essas melhorias se tornam especialmente notáveis ao considerar a aplicação do paradigma de Redes Definidas por Software (SDN) para implementar o balanceamento de carga entre os servidores de conteúdo, agindo diretamente na infraestrutura da rede (nas camadas 2, 3 e 4). Isso é notável porque a maioria dos estudos anteriores se concentrou no balanceamento de tráfego, ao invés de abordar adequadamente o balanceamento de carga. Ainda, o número de trabalhos que utilizam de SDN para realizar balanceamento de carga é elevado porém, são exceções os trabalhos que abordam tal temática com enfoque em distribuição de conteúdo em vídeo.

Além disso, a utilização das métricas de desempenho dos servidores, juntamente com um monitoramento contínuo e a adaptação desses valores conforme a demanda, confere uma dinâmica notável e escalabilidade ao balanceamento de carga. Isso é válido tanto em relação ao aumento do número de clientes quanto à expansão do número de servidores. A principal inovação deste trabalho reside na capacidade da abordagem proposta de gerenciar conexões de grande volume e de longa duração de forma altamente adaptativa. Isso é possível devido à habilidade de manipular conexões, mesmo durante a reprodução de conteúdo multimídia. Em outras palavras, se um usuário estiver consumindo um vídeo e o sistema detectar uma oportunidade para otimizar o balanceamento de carga, essa otimização será realizada de forma transparente, sem que o usuário perceba qualquer alteração em seu conteúdo ou experiência, mesmo durante a migração da conexão.

Tabela 1 – Comparação de Trabalhos Relacionados

Fonte	**Métrica de Desempenho**	**Foco**	**Usa DASH**	**Redirec. Dinâmico**
(LIU; WANG; ZHANG, 2020)	Qualidade de experiência, largura de banda	QoE	Não	Não
(GROUP, 2022)	Localização geográfica e condições de rede	QoE	Sim	Sim
(MAJDABADI; WANG; RAKAI, 2022)	Número de sessões simultâneas, qualidade de streaming	QoE	Sim	Não
(ANDJAMBA; ZODI, 2023)	Distribuição de largura de banda, throughput, latência	QoE	Não	Não
(TAHA, 2023)	Atraso de roteamento, custo do caminho	QoS	Não	Não
(BAMHDI, 2023)	Desempenho do serviço VoD, utilização de recursos de rede	QoE	Não	Não
(BUKHARI; AFAQ; SONG, 2023)	Previsão de recursos, desempenho de transmissão	QoE	Não	Não
(CHIANG; LI, 2024)	Atraso médio de espera	QoS	Não	Não
Este Trabalho	*Throughput*, RAM, Disco, CPU	QoE	Sim	Sim

2.7 RESUMO DO CAPÍTULO

Este capítulo oferece a base teórica essencial para uma compreensão dos sistemas de distribuição de conteúdo e suas potenciais soluções de balanceamento de carga. Dessa forma, é dissertado de maneira introdutória sobre as redes de computadores assim como as redes definidas por software. O capítulo descreve o funcionamento de vídeos adaptativos com enfase no padrão MPEG-DASH e ainda a definição e importância dos codecs. Em seguida, apresenta os métodos de balanceamento de carga mais comuns encontrados em sistemas atuais. Conceitos introdutórios à computação de borda são resumidos. Da mesma forma, o capítulo introduz o conceito de cache e suas principais abordagens. Finalmente, são apresentados trabalhos relacionados destacando as características dos mesmos, apresentando a contribuição deste trabalho.

3 DISCUSSÃO DO PROBLEMA

Este capítulo se dedica a uma análise do desafio de balanceamento de carga em sistemas de distribuição de conteúdo. Inicialmente, são abordados os desafios primordiais associados à gestão das conexões, exemplificados por meio de um estudo de caso do algoritmo Round Robin. Em seguida, é explorado o impacto da geolocalização no que diz respeito à qualidade da experiência do usuário final. Por fim, uma análise aprofundada da relação entre a latência e o *throughput* do sistema é apresentada e discutida em detalhes.

3.1 DESAFIO DE GERENCIAMENTO DE CONEXÕES

O gerenciamento de conexões entre servidores de transmissão de conteúdo é uma tarefa complexa por diversas razões. Diferentemente das conexões voltadas para a navegação em sites da web, que geralmente envolvem a transferência de quantidades de dados predefinidas, as transmissões de conteúdo multimídia são caracterizadas por conexões de fluxo contínuo e volumes significativos de dados. Para ilustrar as complexidades envolvidas, podemos analisar um cenário específico:

1. Algoritmo Round Robin Baseado em DNS: Neste exemplo, o cenário para análise envolve a utilização do algoritmo de balanceamento de carga Round Robin baseado em DNS. Esse algoritmo é amplamente adotado devido à sua simplicidade e eficiência na distribuição de solicitações dos clientes entre os servidores disponíveis.

2. Solicitações de Clientes: há vários clientes que acessam simultaneamente um serviço de streaming de vídeo que utiliza o algoritmo Round Robin para balanceamento de carga (atendimento das solicitações). A ideia inicial é distribuir igualmente as solicitações de reprodução de vídeo entre os servidores disponíveis.

3. Comportamento dos Clientes: Contudo, o comportamento dos clientes não é homogêneo. Alguns clientes podem assistir apenas o início do conteúdo e, em seguida, abandonam a reprodução, enquanto outros optam por assistir o vídeo completo.

4. Desigualdade na Carga dos Servidores: Essa discrepância no comportamento dos clientes leva a um problema significativo. Os servidores que atendem aos clientes que assistem o conteúdo completo ficam sobrecarregados, uma vez que continuam a fornecer o fluxo de vídeo completo, enquanto os servidores que atendem aos clientes que abandonam rapidamente a reprodução ficam com recursos ociosos.

5. Ineficiência na Alocação de Recursos: Essa ineficiência na alocação de recursos demonstra uma das limitações do Round Robin em ambientes de streaming de conteúdo multimídia. A abordagem equitativa não leva em consideração a duração real das conexões ou o

volume de dados transmitidos, o que pode resultar em uma experiência insatisfatória para os usuários.

Em síntese, o desafio de gerenciamento de conexões em ambientes de transmissão de conteúdo multimídia revela-se complexo e multifacetado. O exemplo analisado, utilizando o algoritmo Round Robin baseado em DNS, destaca a dificuldade em equilibrar a carga entre servidores diante do comportamento heterogêneo dos usuários. A busca por uma distribuição equitativa de solicitações, embora eficiente em determinados contextos, revela-se insuficiente quando se trata de otimizar a alocação de recursos em transmissões contínuas de vídeo. A ineficiência na consideração da duração real das conexões e do volume de dados transmitidos destaca uma limitação significativa dessa abordagem. Portanto, é imperativo explorar estratégias mais refinadas e adaptáveis para o gerenciamento de conexões em ambientes de streaming, a fim de proporcionar uma experiência mais satisfatória e eficiente para os usuários.

3.2 GEOLOCALIZAÇÃO

A geolocalização desempenha um papel essencial nas redes de distribuição de conteúdo, e isso decorre da estreita relação entre latência e *throughput* em redes de computadores. A latência, que representa o atraso temporal experimentado pelos dados à medida que atravessam uma rede, desempenha um papel significativo na determinação do *throughput*, que se refere à quantidade de dados que pode ser transmitida em um intervalo de tempo específico. Ainda, é importante notar que a latência tende a aumentar à medida que a distância entre o nó de origem e o nó de destino cresce. Essa conexão entre geolocalização e o desempenho da rede de distribuição de conteúdo se deve à influência direta da distância geográfica sobre a latência, impactando, assim, a qualidade da experiência do usuário. À medida que a distância geográfica se estende, os dados enfrentam um aumento na latência, o que, por sua vez, restringe o *throughput* eficaz, afetando a eficiência da distribuição de conteúdo.

Em termos práticos, é possível discernir diversas razões pelas quais um aumento na latência resulta em uma redução no *throughput* de uma rede:

- Sobrecarga devido a pacotes de controle: Em redes de comunicação, é necessário alocar parte da largura de banda disponível para pacotes de controle, como confirmações de recebimento (ACK) e solicitações de retransmissão. Em redes com elevada latência, os dispositivos devem aguardar períodos mais prolongados para receber esses pacotes de controle, resultando em uma redução do *throughput* efetivo.

- Tempo de ida e volta (*Round-Trip Time* (RTT)): O RTT corresponde ao período temporal necessário para que um pacote de dados viaje do remetente ao destinatário e retorne. Em redes com elevada latência, o RTT é notavelmente ampliado, acarretando um aumento no tempo de espera por confirmações e *feedback*, com consequente redução no *throughput*.

- Janelas de congestão menores: Em cenários de alta latência, as janelas deslizantes do TCP tendem a diminuir, o que implica que o remetente envia uma quantidade menor de dados antes de aguardar por confirmações, culminando em um *throughput* inferior.

- Perda de pacotes: Nas redes com latência elevada, a probabilidade de perda de pacotes devido a colisões, erros de transmissão ou congestionamento é acentuada. Isso gera a necessidade de retransmissões, afetando negativamente o *throughput*.

- Espera por recursos: Redes de alta latência podem requerer um período de espera mais prolongado para que os dispositivos acessem recursos compartilhados, como o meio de transmissão. Tal espera resulta em atrasos na transmissão de dados e, consequentemente, em uma redução no *throughput*.

É válido ressaltar que, em redes com alta latência, o *throughput* pode não estar inteiramente limitada pela latência, dependendo da disponibilidade da largura de banda. No entanto, a latência continua a exercer influência negativa sobre o desempenho da rede, especialmente em cenários de transferência de dados em lotes e aplicações sensíveis à latência, como videoconferência e jogos online. De forma geral, a interação entre latência e *throughput* em redes de computadores é um campo de estudo complexo, sujeito a múltiplos fatores. Em geral, o aumento da latência tende a reduzir o *throughput* devido a atrasos, retransmissões e sobrecargas de controle.

Logo, a compreensão do impacto direto da distância geográfica na latência ressalta a importância dessa variável na determinação da qualidade da experiência do usuário final. Ao explorar as razões práticas pelas quais o aumento na latência resulta em uma redução no *throughput*, desde a sobrecarga devido a pacotes de controle até a espera por recursos, evidencia-se a complexidade desse fenômeno. Diante dessa intrínseca relação, a consideração de estratégias para mitigar os efeitos adversos torna-se crucial. Uma alternativa promissora é a exploração de nós de borda como mecanismos de cache. Ao posicionar esses nós estrategicamente em locais geograficamente dispersos, é possível reduzir significativamente a distância entre os pontos de origem e destino dos dados, mitigando assim os problemas de latência. Além disso, a utilização de nós de borda permite armazenar conteúdo frequentemente acessado em proximidade física aos usuários, otimizando a entrega e, consequentemente, melhorando o *throughput*. Essa abordagem não apenas endereça os desafios inerentes à sobrecarga devido a pacotes de controle, tempo de ida e volta prolongado (RTT), janelas de congestão menores e perda de pacotes, mas também oferece uma solução proativa para garantir uma experiência do usuário mais eficiente em cenários sensíveis à latência. Em resumo, a implementação estratégica de nós de borda como cache emerge como uma solução promissora para atenuar os impactos negativos da geolocalização na latência e, por conseguinte, no *throughput*, contribuindo para o aprimoramento global do desempenho das redes de distribuição de conteúdo.

3.3 RESUMO DO CAPÍTULO

Esse capítulo, aborda de maneira abrangente os desafios associados ao balanceamento de carga em sistemas de distribuição de conteúdo, com foco especial no gerenciamento de conexões e na influência da geolocalização. A análise do desafio de gerenciamento de conexões destacou a complexidade envolvida na distribuição equitativa de solicitações em ambientes de transmissão de conteúdo multimídia, evidenciando as limitações do algoritmo Round Robin baseado em DNS. A heterogeneidade no comportamento dos usuários e a necessidade de considerar a duração real das conexões e o volume de dados transmitidos foram apontadas como elementos cruciais a serem abordados para otimizar a alocação de recursos. Além disso, a discussão sobre a geolocalização ressaltou a relação direta entre latência e *throughput*, com uma análise detalhada das razões práticas pelas quais o aumento na latência impacta negativamente a eficiência da rede.

4 PROPOSTA

Haja visto os principais desafios e complexidades apresentadas nas seções anteriores, este capítulo tem como propósito apresentar a solução proposta. Primeiramente é apresentada a abordagem utilizada no balanceamento de carga, quais as métricas utilizadas e motivação para tal. Em seguida, é destacada a importância das redes definidas por software na elaboração da abordagem. Depois disso, as políticas de cache utilizadas são discorridas e por último é apresentada e discutida a arquitetura geral do sistema assim como a utilização de nós na borda para otimização da qualidade de experiência do usuário.

4.1 ABORDAGEM DE BALANCEAMENTO DE CARGA

A estratégia adotada para mitigar o desafio da sobrecarga de trabalho em servidores de conteúdo multimídia é baseada em um monitoramento contínuo dos recursos disponíveis, com o objetivo de realizar o redirecionamento de tráfego para o servidor mais adequado para estabelecer novas conexões. O monitoramento de recursos é uma prática amplamente consagrada, e quando integrada a outras medidas, desempenha um papel crítico na manutenção da eficiência do sistema. Esse processo é ininterrupto, uma vez que a verificação regular da utilização dos recursos é imperativa para o bom funcionamento do sistema.

Apesar da complexidade associada ao redirecionamento de tráfego para o servidor mais apropriado, uma estratégia ainda pouco explorada, pois envolve a manipulação de fluxos durante a reprodução do conteúdo, as chances de erros, atrasos e artefatos na reprodução são substanciais. No entanto, o procedimento proposto nesta pesquisa acontece de forma imperceptível, o que o torna singular em relação às abordagens encontradas na literatura. Essa característica distintiva reforça a contribuição única deste trabalho.

4.2 TCP HANDOFF

A técnica de transferência de conexão TCP (*TCP Handoff*) já é conhecida no campo das redes de comunicação pois desempenha um papel fundamental na gestão eficiente de conexões em ambientes distribuídos e altamente dinâmicos. Isso porque tal técnica permite que uma conexão ativa seja migrada de um servidor para outro sem interromper a sessão do usuário (KIM; RIXNER, 2006). Contudo, quando aplicada a uma infraestrutura legada, seu emprego se mostra altamente complexo, demandando recursos de hardware e tecnologias específicas que, geralmente, não fazem parte da infraestrutura já estabelecida na rede. Esta complexidade e a necessidade de recursos adicionais frequentemente tornam a adoção da técnica desafiadora em ambientes de rede legada (TANENBAUM; STEEN, 2001).

No entanto, com a ascensão das redes definidas por software (SDN), abre-se uma nova perspectiva. A técnica pode ser implementada de maneira menos complexa e ainda mais eficiente em ambientes SDN. Isso se deve ao fato de que, nas redes SDN, o controle de fluxos de conexões

é mais granular, tornando a manipulação de pacotes mais ágil e precisa. Isso representa um avanço significativo, pois permite a integração da técnica de transferência de conexão TCP de forma mais harmoniosa em ambientes de rede modernos, abrindo possibilidades para melhorias substanciais na gestão de conexões e na otimização do desempenho da rede. Este avanço exemplifica o potencial transformador das tecnologias de rede definidas por software na resolução de desafios anteriormente enfrentados em redes legadas.

A introdução da técnica abordada neste estudo começa quando um cliente faz sua primeira requisição. A partir desse momento, inicia-se o processo de seleção do servidor mais apropriado para atender às necessidades do cliente. No entanto, a fim de evitar transições frequentes de servidor por parte do cliente, o controlador registra cada alteração e estabelece um período de espera mínimo antes de autorizar uma nova troca. Esse intervalo de tempo permite que o cliente se adapte e se estabilize no novo ambiente do servidor. Isso, por sua vez, previne que clientes em situações desfavoráveis realizem mudanças de servidor de forma indiscriminada, contribuindo para uma experiência mais consistente e eficiente.

Para migrar a sessão TCP ativa de um cliente, o controlador executa o procedimento de exclusão dos dois fluxos correspondentes bloqueando a comunicação entre o antigo servidor e o cliente. Como resultado dessa exclusão, o próximo pacote enviado pelo cliente é redirecionado para o controlador da rede SDN.

Novos fluxos são então estabelecidos entre o cliente e o novo servidor por parte do controlador destacando que o novo servidor não reconhecerá a sessão do cliente, levando-o a enviar um pacote *Reset* (RST) como resposta, forçando o cliente a reiniciar a sessão TCP por meio de um pacote *Synchronous* (SYN), ao qual o novo servidor responde com um pacote SYN+*Acknowledgement* (ACK).

A partir deste ponto, o cliente está apto a solicitar os segmentos de vídeo necessários ao novo servidor. Em seguida, o controlador é capaz de forjar dois novos pacotes com base nas informações de sequência e reconhecimento extraídas do pacote do cliente. Esses pacotes simulam a intenção do cliente de encerrar a sessão: *Finalize* (FIN) e ACK. O primeiro informa ao antigo servidor a intenção do cliente de encerrar a sessão, e o servidor responde com um pacote FIN+ACK.

O controlador pode gerar esses dois pacotes mesmo sem a resposta da origem antiga, enviando o primeiro pacote e aguardando um curto intervalo de dois milissegundos antes de enviar o pacote ACK subsequente, encerrando assim a sessão TCP antiga de maneira apropriada. Esse processo demonstra a habilidade do controlador em gerenciar e orquestrar a transição de sessões TCP com precisão e eficiência.

4.3 MÉTRICAS MONITORADAS

Ao manter uma vigilância constante sobre o uso de recursos pelos servidores, é possível assegurar que eles estejam operando dentro ou próximos aos seus limites de capacidade. Quando

os servidores atingem seu limite, podem tornar-se lentos e ineficientes, prejudicando o desempenho global do sistema. O monitoramento periódico capacita a detecção precoce de quando é necessário adotar medidas preventivas, como alocar recursos adicionais ou, no contexto da estratégia proposta neste estudo, redistribuir a carga de trabalho.

Além disso, essa abordagem não só contribui para a otimização do desempenho, mas também minimiza o risco de interrupções não planejadas, garantindo a continuidade do serviço. A análise constante dos recursos utilizados pelos servidores possibilita uma tomada de decisão informada e eficaz, contribuindo assim para a estabilidade e aprimoramento contínuo do sistema.

Um dos desafios inerentes a esse tipo de abordagem está relacionado à seleção apropriada das métricas utilizadas para a seleção do mais adequado servidor de conteúdo para o balanceamento da carga. A escolha das métricas adequadas impacta diretamente o desempenho final do sistema de balanceamento de carga. Assim, a seleção precisa das métricas é fundamental, uma vez que essas métricas servem como indicadores-chave para avaliar o estado do sistema e tomar decisões embasadas. Métricas inadequadas podem levar a avaliações imprecisas e, consequentemente, à tomada de decisões equivocadas, comprometendo a eficácia do sistema de balanceamento de carga. Portanto, as seguintes métricas foram escolhidas: *Throughput*, Consumo de CPU, Carga de Memória, Utilização do sistema de armazenamento e Latência de Comunicação dos servidores. A seguir é discorrido sobre cada uma das métricas e suas respectivas influências no *streaming* de vídeo.

4.3.1 *Throughput*

A mensuração do *throughput* desempenha um papel crucial na identificação e resolução de gargalos no contexto de infraestrutura de servidores. Quando o *throughput* se situa abaixo de um limiar preestabelecido isso serve como um indicativo potencial de sobrecarga do servidor. Essa sobrecarga pode ser desencadeada por diversos fatores, tais como um aumento súbito e significativo no tráfego, ou mesmo por configurações inadequadas do servidor em si. Portanto, a análise do *throughput* se revela uma ferramenta fundamental para monitorar e otimizar o desempenho do servidor, garantindo assim a qualidade e a disponibilidade dos serviços prestados.

Outro ponto de relevância crítica no monitoramento dessa métrica é o potencial para aprimorar significativamente a qualidade da experiência do usuário. Isso se traduz em uma capacidade ampliada de transmitir conteúdo multimídia em alta qualidade, livre de artefatos visuais e interrupções, resultando em uma experiência mais satisfatória para o usuário.

Para ilustrar essa questão, considera-se um cenário no qual o *throughput* da rede pública de um servidor em nuvem é de 10 *Megabits Per Second* (Mbps) e a taxa de bits média necessária para cada transmissão é de 1 Mbps. Nesse contexto, é possível observar que o sistema entra em sobrecarga quando há 10 clientes simultâneos utilizando a plataforma. No entanto, a situação se agrava substancialmente quando o número de clientes aumenta para 100.

Com 100 clientes ativos, cada cliente tem acesso a uma largura de banda limitada a aproximadamente 100 *Kilobits Per Second* (Kbps) para receber os dados. Essa restrição severa

na largura de banda disponível resulta em diversos problemas, incluindo latência significativa e atrasos perceptíveis no fluxo de dados.

4.3.2 Consumo de CPU

O monitoramento do uso da Unidade Central de Processamento (*Central Process Unit* (CPU)) é relevante, uma vez que reflete a capacidade do servidor de processar novas tarefas. Quando a CPU se encontra sobrecarregada, isso implica que o servidor não consegue atender a um grande número de clientes, desencadeando uma série de eventos adversos que podem afetar negativamente o desempenho da rede. Como resultado, os clientes podem enfrentar dificuldades na obtenção dos dados de que necessitam, resultando em um aumento na latência e no atraso da comunicação. Esses problemas podem, por sua vez, aumentar ainda mais a carga sobre a CPU, levando, em última instância, à paralisação do servidor.

4.3.3 Carga de Memória - RAM

A memória *Random Access Memory* (RAM) desempenha um papel fundamental na reprodução de vídeos, embora seu consumo seja geralmente menor em comparação com outras tarefas. No entanto, sua monitorização é de extrema importância, uma vez que o seu esgotamento pode acarretar uma série de problemas significativos. Um dos problemas mais críticos é a possibilidade de encerramento forçado de processos de serviço quando a memória atinge um limite crítico, o que, por sua vez, resulta na recusa de novas conexões.

Além disso, existe a possibilidade de que a falta de memória RAM afete significativamente a qualidade da reprodução de vídeos. Nesse caso, quando a RAM está quase esgotada, o sistema pode começar a utilizar a memória virtual (espaço em disco) como uma extensão da RAM. No entanto, essa troca de dados entre a RAM e a memória virtual é significativamente mais lenta, o que pode resultar em atrasos, pausas indesejadas e até mesmo travamentos durante a reprodução de vídeos. Isso é especialmente crítico em situações em que a reprodução de vídeo envolve conteúdo de alta definição, onde a latência e a fluidez são essenciais.

Dessa forma, a falta de memória RAM pode impactar negativamente a qualidade da reprodução de vídeos, causando atrasos, travamentos e interrupções no fluxo da mídia, o que, por sua vez, afeta a experiência do usuário. Portanto, a monitorização e a gestão eficaz da memória RAM são cruciais para garantir uma reprodução de vídeo suave e ininterrupta.

4.3.4 Disco

O subsistema de Entrada/Saída (E/S), notadamente o dispositivo de armazenamento desempenha um papel de extrema importância para assegurar a operação ininterrupta e eficiente de sistemas de *streaming* de vídeo, uma vez que o conteúdo de vídeo, juntamente com seus metadados e ativos correlatos, é tipicamente armazenado em dispositivos de armazenamento em

disco. A E/S do dispositivo é encarregada da leitura e escrita de dados de e para esses dispositivos, exercendo um impacto direto na capacidade do servidor em atender às solicitações dos usuários.

Quando um usuário requisita a reprodução de um vídeo, o servidor é encarregado de acessar o arquivo de vídeo no dispositivo de armazenamento não volátil e disponibilizá-lo para *streaming* de forma eficaz. A eficiência das operações de E/S é, portanto, um fator crítico para a recuperação ágil do conteúdo e para a minimização da latência.

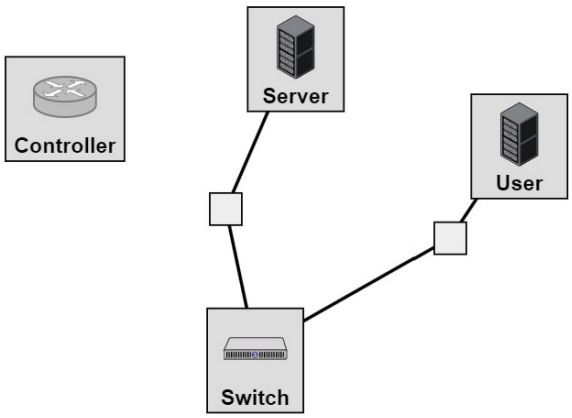
É importante ressaltar que, caso a utilização desse recurso ultrapasse um determinado limiar predefinido, a experiência do usuário pode ser significativamente afetada, uma vez que a ocorrência de atrasos torna-se provável, levando a interrupções no carregamento do *buffer* de vídeo durante a reprodução da mídia. Isso pode resultar em uma experiência de visualização insatisfatória, com impactos negativos na satisfação do usuário e na qualidade do serviço de *streaming* prestado. Portanto, a otimização da gestão das operações de E/S de disco é fundamental para garantir uma experiência de *streaming* de vídeo fluida e de alta qualidade.

4.4 DEFINIÇÃO DOS LIMITES DE UTILIZAÇÃO

A escolha do servidor mais adequado a receber a conexão é baseado num sistema de ranqueamento simples no qual cada métrica recebe uma pontuação baseada no quão seus maus valores podem ser prejudiciais ao desempenho da QoE final. Para obter qual métrica possui mais influência sob a qualidade de experiência do usuário, testes de estresse foram realizados.

Através da ferramenta *stress* (UBUNTU, 2019), todos as métricas analisadas foram colocadas sob máxima utilização ou estresse. A realização dos testes utilizou da topologia simples conforme a Figura 3. Todos os comandos foram executados no servidor.

Figura 3 – Topologia para teste de estresse



Fonte: O autor, 2024

4.4.1 CPU

O objetivo é colocar todos os núcleos da CPU sob carga total. Isso permite testar a resposta do sistema sob condições de elevada carga. O seguinte comando foi utilizado no servidor:

```
stress --cpu \$(nproc)
```

De modo geral tal comando descobre quantos núcleos de CPU o sistema possui, inicia um número de processos igual ao número de núcleos de CPU e ainda cada processo executa cálculos intensivos para manter todos os núcleos da CPU ocupados.

4.4.2 RAM

A finalidade do experimento de estresse para o consumo de memória RAM é criar o máximo de carga de memória no sistema. O seguinte comando foi utilizado no servidor:

```
stress --vm 4 --vm-bytes 220M
```

O comando inicia 4 processos onde cada trabalhador aloca 220 MB de memória e ainda cada processo exercita a memória alocada, escrevendo nela continuamente.

4.4.3 Disco

Assim como nas métricas anteriores o objetivo é colocar o máximo de carga no servidor referente a escrita e leitura do dispositivo. O seguinte comando foi utilizado no servidor:

```
stress --io 10
```

Esse comando inicia 10 processos (workers) que geram operações de I/O onde cada processo realiza operações de escrita e leitura no disco repetidamente.

4.4.4 Throughput

Para o teste de *throughput* foi idealizado um cenário onde o servidor é sobrecarregado com novas conexões a serem gerenciadas. Dessa forma, o seguinte comando foi executado no servidor:

```
watch -n 1 ab -n 10000 -c 1000 -k [address]
```

O comando *watch -n 1* executa o código *ab -n 10000 -c 1000 -k [address]* a cada segundo. Ou seja, a cada segundo, o *ab* (*ApacheBench* (FOUNDATION, 2024)) enviará 10.000 solicitações ao servidor local ([address]) para o recurso /index.html, com 1.000 dessas solicitações sendo feitas simultaneamente e utilizando conexões *KeepAlive*.

4.4.5 Resultados

Foram conduzidos um total de 10 execuções para cada experimento sendo que a média dessas execuções foi adotada como o valor final. Cada teste foi realizado em condições con-

troladas, com a reinicialização do controlador, dos servidores e do *cache* do navegador a cada iteração, garantindo assim a consistência e confiabilidade dos resultados.

As métricas adotadas para a análise dos testes realizados estão intrinsecamente relacionadas à mensuração da QoE do cliente. Essas métricas abrangem diferentes aspectos que impactam diretamente na percepção e satisfação do usuário final:

- Média do *Bitrate* por segundo: Esta métrica destaca a qualidade média do vídeo ao longo do tempo. O *bitrate* é um indicador crucial, e sua média proporciona *insights* sobre a qualidade geral da experiência de visualização. Foram gerados gráficos distintos para cada *dataset*.

- Número de mudanças na qualidade do vídeo: Quanto menor o número de mudanças, melhor a estabilidade e qualidade da experiência do usuário. Esta métrica reflete a consistência da transmissão, um fator determinante para a satisfação do cliente.

- Tempo de carregamento da página: O tempo necessário para carregar a página é um indicador crucial da eficiência do serviço. Menores tempos de carregamento contribuem significativamente para uma experiência mais fluida e satisfatória.

- Tempo para a exibição do primeiro quadro do vídeo no *player*: Este intervalo de tempo representa o início efetivo da experiência de visualização. Quanto menor o tempo para a exibição do primeiro quadro, mais rápida e envolvente é a entrada do usuário no conteúdo.

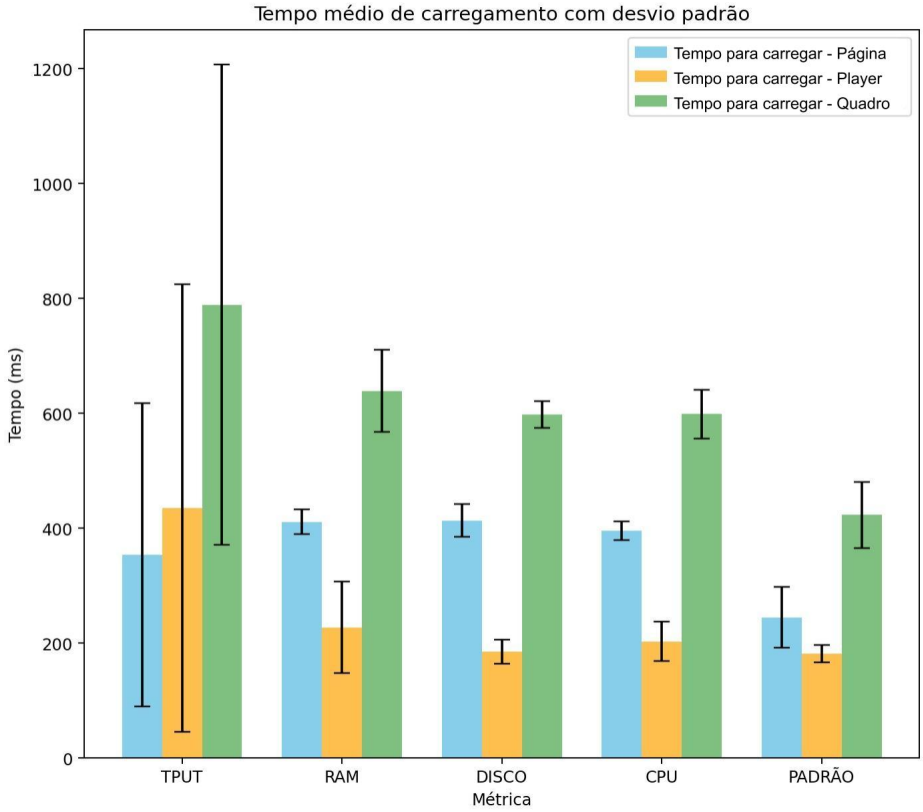
- Número de *stalls* no vídeo: *Stalls*, ou interrupções na reprodução do vídeo, são fatores que podem prejudicar a experiência do usuário. O objetivo é minimizar ou, idealmente, eliminar essas interrupções para proporcionar uma experiência de visualização ininterrupta e agradável.

A Figura 4 ilustra o tempo de resposta conforme cada teste de estresse era executado. Todos os testes foram realizados 10 vezes, e a média e o desvio padrão foram utilizados na elaboração do gráfico. Além disso, os resultados dos testes sem influência de estresse foram computados representados no gráfico como *"PADRÃO"*, ou seja, nenhum outro aplicativo ou processo custoso estava em execução durantes os testes, somente os necessários para o funcionamento do sistema operacional.

Claramente, a métrica que mais influenciou o tempo de resposta foi o *throughput*. Como o objetivo do teste era maximizar a carga de trabalho do servidor, o número de conexões e de conexões simultâneas teve que ser significativamente aumentado. Em determinados momentos, a espera para o consumo do conteúdo era visivelmente perceptível e por isso seu desvio padrão foi elevado.

Em relação às outras métricas, observou-se uma deterioração em comparação com os resultados padrão, ou seja, sem estresse. No entanto, essa deterioração não foi significativa a ponto de causar uma mudança perceptível na qualidade da experiência do usuário.

Figura 4 – Resultados do teste de stress - tempo de resposta



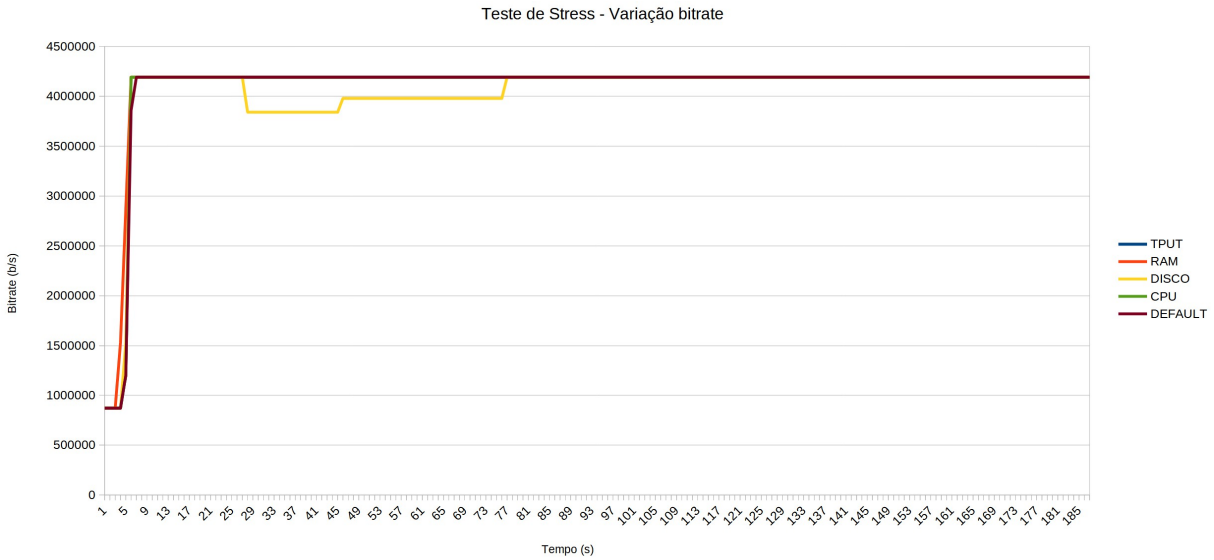
Fonte: O autor, 2024

A Figura 5 apresenta a variação do *bitrate* durante a reprodução do conteúdo conforme estressava-se cada uma das métricas de desempenho do servidor. Observa-se que não houve impactos significativos, exceto no teste realizado com o disco sob estresse. Por volta do segundo 28, nota-se um leve declínio no *bitrate*, que só retorna à normalidade por volta do segundo 80. A degradação na experiência do usuário causada pelo disco sob estresse pode ser explicada pelo fato de que, quando o usuário requisita novos segmentos de vídeo, o servidor pode não conseguir recuperar e enviar esses segmentos em tempo hábil pois é natural que a latência de leitura aumente e com isso o tempo necessário para acessar e ler os dados do disco se prolonga, resultando em um atraso na disponibilização dos segmentos de vídeo. No momento dos testes, nenhuma das métricas de desempenho avaliadas apresentou *stall*.

De forma a avaliar se a execução dos testes de estresse para cada uma das métricas refletido no tempo de carregamento e *bitrate* produziram resultados significativos, estatisticamente falando, foram realizados testes estatísticos sobre esses resultados.

A técnica de Kruskal-Wallis (KRUSKAL; WALLIS, 1952) é um método não paramétrico utilizado para verificar se há diferenças significativas entre dois ou mais grupos de dados. Seu funcionamento se baseia em atribuições e classificações aos seus pontos de dados, independentemente de seus valores reais, e posteriormente há uma análise dessas classificações para ver se há um padrão. Se os dados forem realmente idênticos entre os grupos, as classificações deverão

Figura 5 – Resultados do teste de stress - Variação do *bitrate*



Fonte: O autor, 2024

ser embaralhadas de forma bastante aleatória e é gerado um valor p. Um valor p baixo (inferior a 0.05, ou seja, com um grau de confiança de 95%) indica que os dados são provavelmente diferentes em pelo menos alguns dos grupos. Este método foi escolhido pois não há pré requisitos para sua utilização como o ANOVA (FISHER, 1925) que exige certas características como uma distribuição normal dos dados por exemplo.

Dessa forma, referente ao tempo de carregamento total, comparando-se as métricas obtém-se o valor p igual a 0.0081 e este valor p é menor que o nível de significância comum de 0.05. Logo, rejeita-se a hipótese nula, indicando evidências significativas de que existem diferenças entre as métricas avaliadas.

Os resultados da variação do *bitrate* também passaram pelo teste de Kruskal-Wallis. O resultado gerou um valor p de 2.85×10^{-54}, muito inferior ao nível de significância comum de 0.05. Por isso, rejeita-se a hipótese nula, indicando evidências fortes de que existem diferenças significativas entre as métricas avaliadas.

No seguimento das implicações negativas associadas à utilização excessiva das métricas selecionadas, foram propostas as seguintes diretrizes iniciais para redirecionamento de conexões, a fim de selecionar o servidor mais apropriado:

- Com o maior *throughput*, atribui-se 2 pontos.
- Com a menor utilização da CPU, atribui-se 1 ponto.
- Com a menor utilização da RAM, 1.
- O servidor com a menor utilização do disco, 3 pontos.

Defina a função de pontuação Score(s) para um determinado servidor s da seguinte forma:

$$\text{Score}(s) = \sum_{i=1}^{4} \cdot \text{Metric}_i(s) \tag{1}$$

Onde:

$$\text{Metric}_1(s) = \begin{cases} 2 & \text{if Throughput}(s) = \max(\text{Throughput}) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Metric}_2(s) = \begin{cases} 1 & \text{if CPU\_Usage}(s) = \min(\text{CPU\_Usage}) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Metric}_3(s) = \begin{cases} 1 & \text{if RAM\_Usage}(s) = \min(\text{RAM\_Usage}) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Metric}_4(s) = \begin{cases} 3 & \text{if Disk\_Usage}(s) = \min(\text{Disk\_Usage}) \\ 0 & \text{otherwise} \end{cases}$$

Onde *Metric* representa o peso atribuído a cada métrica. O Algoritmo 1 ilustra a execução do sistema em forma de pseudo-código.

O algoritmo é detalhado a seguir:

- Inicialização: O algoritmo começa inicializando parâmetros, incluindo o conjunto de servidores, intervalo de monitoramento e limiares de conexão.
- Laço de Monitoramento (Primeiro *While Loop*): O primeiro *loop while* é executado indefinidamente e, em cada iteração, percorre cada servidor no conjunto para monitorar suas métricas. A função *MonitorMetrics* coleta informações sobre o uso da CPU, uso da RAM, *throughput* e uso do disco para cada servidor.
- Função de Monitoramento de Métricas (*MonitorMetrics*): Esta função obtém várias métricas para um determinado servidor e atualiza os atributos do servidor com os valores obtidos.
- Função de Seleção de Servidor (*ChooseBestServer*): Esta função itera pelo conjunto de servidores, calcula uma pontuação para cada servidor usando a função *CalculateScore* e seleciona o servidor com a pontuação mais alta como o mais adequado servidor.
- Função de Cálculo de Pontuação (*CalculateScore*): Esta função calcula uma pontuação para um servidor com base em sua *throughput*, uso de CPU, uso de RAM e uso de disco. As pontuações são baseadas em critérios específicos, como atribuir pontuações mais altas para um *throughput* mais alto, uso mais baixo de CPU, RAM e disco, conforme Equação 1.

Algorithm 1: *Load Balancing System with Tie-Breaking*

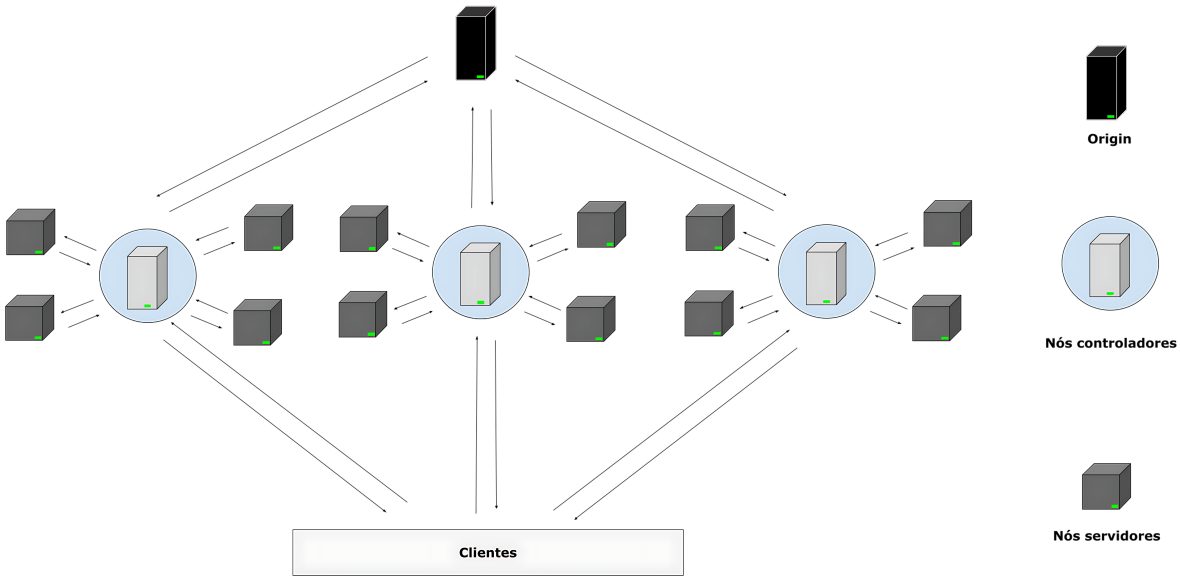
```
Data: Server pool, Monitoring interval, Connection thresholds
Result: Redirect connections to the appropriate server
while True do
    for each server in server pool do
        MonitorMetrics(server);
    selectedServer ← ChooseBestServer(server pool);
    RedirectConnection(selectedServer);
    Sleep for monitoring interval;
Function MonitorMetrics(server)
    server.cpu_usage ← GetCPUUsage(server);
    server.ram_usage ← GetRAMUsage(server);
    server.throughput ← GetThroughput(server);
    server.disk_usage ← GetDiskUsage(server);
Function ChooseBestServer(serverPool)
    maxScore ← 0;
    bestServers ← [];
    for each server in serverPool do
        score ← CalculateScore(server);
        if score > maxScore then
            maxScore ← score;
            bestServers ← [server];
        else if score = maxScore then
            bestServers.append(server);
    if length(bestServers) = 1 then
        return bestServers[0];
    else
        // Tie-break using throughput and random choice bestServer ← TieBreak(bestServers);
        return bestServer;
Function TieBreak(servers)
    // Sort servers by throughput (descending) servers.sort(key=lambda x: (x.throughput),
      reverse=True);
    return servers[0];
    // Return the server with the best throughput
Function CalculateScore(server)
    score ← 0;
    if max(server.throughput) then score + = score + 2;
    // 2 point for highest throughput if min(server.cpu_usage) then score + = score + 1;
    // 1 point for lowest CPU usage if min(server.ram_usage) then score + = score + 1;
    // 1 point for lowest RAM usage if min(server.disk_usage) then score + = score + 3 ;
    // 3 point for lowest disk usage return score;
Function RedirectConnection(selected_server)
    Log("Redirecting connection to", selected_server);
```

- Função de Desempate (*TieBreak*): Esta função tenta romper o empate considerando o *throughput* (prioridade para servidores com maior *throughput*). Se ainda houver empate após essa consideração, a escolha é feita aleatoriamente entre os servidores empatados.
- Função de Redirecionamento de Conexão (*RedirectConnection*): Uma vez escolhido o melhor servidor, o sistema redireciona a conexão recebida para esse servidor.

4.5 ARQUITETURA DO SISTEMA DE BALANCEAMENTO DE CARGA

A arquitetura proposta para o sistema de balanceamento de carga é concebida de maneira hierárquica, apresentando três níveis. Essa abordagem confere maior robustez ao sistema, uma vez que, ao agrupar recursos em estruturas hierárquicas, facilita o gerenciamento e a distribuição da carga de trabalho, mesmo à medida que o sistema cresce e os recursos se dispersam por diferentes localizações geográficas. A Figura 6 ilustra os três níveis hierárquicos propostos.

Figura 6 – Arquitetura do Sistema de Balanceamento de Carga



Fonte: O autor, 2023

4.5.1 Origem

Na camada inicial, denominada "Origem", são encontrados os nós de origem. Os servidores de origem desempenham um papel crítico na arquitetura de um serviço de *streaming*. Esses servidores são os repositórios centrais que armazenam todo o vasto catálogo de mídia disponível para os usuários. Eles são caracterizados por uma notável capacidade computacional, o que lhes permite armazenar e gerenciar um grande volume de conteúdo, como filmes, séries, músicas, e outros tipos de mídia.

No entanto, a característica distintiva dos servidores de origem é a sua localização física. Eles geralmente estão posicionados em centros de dados ou locais estratégicos. Essa distância geográfica pode ser significativa e, por vezes, abranger regiões inteiras ou até mesmo países. Isso ocorre por razões de eficiência e escalabilidade da infraestrutura. Como resultado, a decisão de redirecionar a conexão para um nó de origem só é efetivada quando nenhum dos nós servidores possui o conteúdo solicitado pelo usuário em seu repositório local.

4.5.2 Nós Controladores

Os nós controladores desempenham um papel crucial pois são responsáveis por receber a requisição inicial dos usuários e atuam como o "cérebro"do sistema. Dentro desses nós, residem os controladores de Redes Definidas por Software (SDN), que têm a função de determinar quais servidores estão mais adequados para receber a conexão do usuário.

Portanto, quando uma requisição é recebida, o controlador inicia o processo de seleção do mais adequado servidor. Uma vez que o servidor ideal é identificado, o controlador inicia o processo de redirecionamento da conexão, direcionando-a para esse servidor específico.

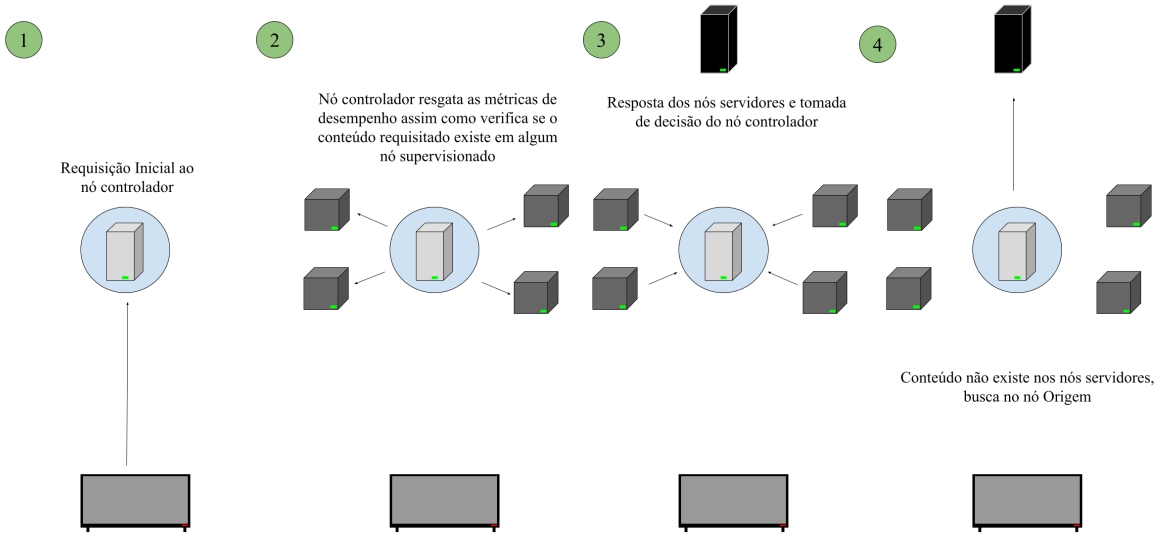
Essa abordagem estratégica garante que as conexões dos usuários sejam otimizadas, direcionadas para os servidores mais adequados, e, consequentemente, resulta em um desempenho superior do sistema como um todo.

4.5.3 Nós Servidores

Os nós servidores são nós de computação de borda e são parte essencial do sistema. Sua localização estratégica os posiciona em proximidade com os usuários finais, a fim de otimizar o desempenho e a latência das comunicações. No entanto, em contrapartida, esses nós de computação de borda possuem recursos computacionais limitados em comparação com servidores de origem.

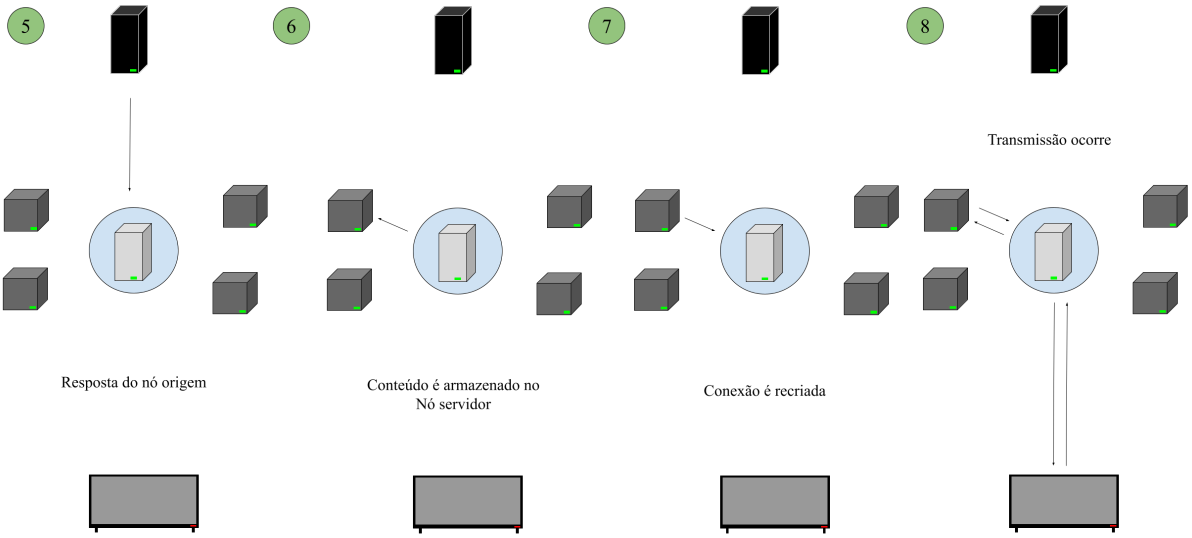
Quando um usuário faz uma requisição, o controlador identifica o servidor mais adequado. Com o monitoramento constante do grupo de servidores, busca identificar nós servidores que contenham o conteúdo requisitado. Após identificar potenciais candidatos, o controlador realiza uma avaliação cuidadosa para determinar o nó mais adequado a receber a conexão do usuário com base nos critérios de utilização do *throughput*, RAM, CPU, disco e latência.

Figura 7 – Funcionamento do balanceamento de carga parte 1



Fonte: O autor, 2023

Figura 8 – Funcionamento do balanceamento de carga parte 2



Fonte: O autor, 2023

4.6 RESUMO DO CAPÍTULO

A solução proposta neste capítulo oferece uma abordagem alternativa para o balanceamento de carga em servidores de conteúdo multimídia. A estratégia adotada é baseada em um monitoramento contínuo dos recursos disponíveis, permitindo o redirecionamento de tráfego para o servidor mais apropriado. Destaca-se a importância das redes definidas por software (SDN) na implementação eficiente dessa abordagem, especialmente ao empregar a técnica de transferência de conexão TCP (TCP *Handoff*). Apesar da complexidade envolvida, a singularidade da

abordagem proposta reside na sua execução imperceptível durante a reprodução do conteúdo, minimizando erros e artefatos. As métricas monitoradas, como *Throughput*, utilização de CPU, carga de Memória e utilização do sistema de armazenamento desempenham um papel crucial na manutenção da eficiência do sistema, contribuindo para a otimização do desempenho e prevenção de interrupções não planejadas. Por fim, a arquitetura hierárquica do sistema de balanceamento de carga, composta por nós de origem, nós controladores e nós servidores, proporciona robustez e eficiência, promovendo uma experiência consistente e de alta qualidade para os usuários finais.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo, apresenta-se a metodologia de pesquisa adotada para o desenvolvimento dos experimentos de avaliação da abordagem, incluindo as tecnologias envolvidas, a descrição dos experimentos realizados e a análise dos resultados obtidos.

5.1 EXPERIMENTOS E METODOLOGIA DE AVALIAÇÃO

Nesta seção, são apresentadas as tecnologias utilizadas que possibilitaram a realização dos experimentos assim como a metodologia de avaliação escolhida. Para melhor compreensão do trabalho como um todo, é importante que o leitor tenha familiaridade com algumas aplicações dessas tecnologias.

5.1.1 CloudLab

O CloudLab (CLOUDLAB, 2023) representa uma infraestrutura científica altamente flexível e escalável, concebida para viabilizar a pesquisa científica fundamental na nuvem, sendo uma iniciativa construída e mantida pela própria comunidade científica. Essencialmente, ele funciona como uma "meta-nuvem", permitindo a criação de ambientes de nuvem personalizados e, ao mesmo tempo, oferecendo a capacidade de particionamento para a realização de múltiplos experimentos de forma isolada e simultânea. Esta plataforma é uma contribuição significativa para a pesquisa avançada e oferece recursos para a comunidade científica explorar e avançar em diversos campos do conhecimento. Assim, a utilização do Cloudlab permite a simulação de um cenário de rede distribuída, essencial para testar a eficiência e a robustez do sistema de balanceamento de carga proposto, sob condições diversas de latência e largura de banda.

A atual implantação do CloudLab engloba mais de 25000 núcleos distribuídos em três localizações distintas: na Universidade de Wisconsin, na Universidade Clemson e na Universidade de Utah, todos nos Estados Unidos da América. Essa infraestrutura de computação permite suportar uma ampla variedade de experimentos e projetos de pesquisa de forma distribuída e simultânea.

A decisão de utilizar o CloudLab como plataforma de pesquisa foi motivada por diversos fatores essenciais. Primeiramente, a abrangente distribuição geográfica dos nós do em todo o território americano atende os requisitos de distribuição de nós na infraestrutura. Além disso, a flexibilidade e a capacidade de personalização que o CloudLab proporcionou foram fundamentais na escolha, permitindo adaptar e configurar ambientes de acordo com as especificidades deste estudo. Outro ponto que merece destaque é a liberdade de escolha oferecida em relação à pilha de software, configurações de rede, hardware e outros parâmetros. Essa versatilidade permitiu uma adaptação precisa às necessidades específicas da pesquisa, tornando a plataforma uma escolha lógica e vantajosa para a condução deste trabalho.

5.1.2 Open vSwitch

Uma aplicação de suma importância no contexto deste estudo é o Open vSwitch. O Open vSwitch é um *switch* virtual que oferece suporte ao protocolo OpenFlow, sendo concebido com o propósito de automatizar a gestão de redes de grande escala por meio de extensibilidade programática. Além disso, ele apresenta compatibilidade com diversos protocolos e interfaces de rede (OVS, 2018). A escolha pelo Open vSwitch se fundamenta em sua habilidade de suportar o protocolo OpenFlow e no fato de ser amplamente reconhecido na literatura como o *switch* virtual mais utilizado na atualidade.

5.1.3 Reprodutor de Vídeo MPEG-DASH

O *dash.js* é um reprodutor de vídeo desenvolvido em JavaScript que oferece suporte a arquivos de vídeo no formato MPEG-DASH (DASH.JS, 2012).

De acordo com a pesquisa conduzida por (SILHAVY et al., 2022), o *dash.js* é amplamente utilizado para pesquisas acadêmicas e industriais devido à sua robustez e confiabilidade. Essa ampla adoção significa que o reprodutor tem sido testado em diversos cenários, o que contribui para a sua estabilidade e eficiência. A presença de uma grande comunidade de usuários e desenvolvedores também garante um suporte contínuo e a rápida resolução de possíveis problemas. Além disso, o *dash.js* é interoperável com uma ampla gama de formatos de mídia e tecnologias de *streaming*, facilitando a integração com diferentes sistemas e plataformas.

Outra característica que auxiliou na escolha desse reprodutor foi por ele ser de código fonte aberto. Isso facilita consideravelmente quaisquer modificações necessárias. Além disso, o site de desenvolvimento do software oferece uma ampla gama de funções na biblioteca do aplicativo. Entre as funções já disponíveis para os desenvolvedores, incluem-se a capacidade de recuperar o *bitrate* atual do vídeo e áudio, monitorar a quantidade de *stalls* durante a reprodução de vídeo e identificar as qualidades disponíveis no servidor de conteúdo, entre outras. Isso significa que essa ferramenta não apenas proporciona uma experiência de reprodução de vídeo de alta qualidade, mas também possibilita a monitorização em tempo real da transmissão, permitindo tomadas de decisão que visam a melhoria da QoE final e a otimização eficaz dos recursos disponíveis na CDN.

5.1.4 Conteúdo Multimídia Utilizado nos Experimentos

O *dataset*, denominado ***CAR CENC*** (YOUTUBE, 2012), consiste de um vídeo com duração de 181 segundos, utilizando compressão x264 disponível em seis diferentes resoluções (qualidades) de vídeo, acompanhados por uma faixa de áudio, conforme descrito a seguir:

- **Áudio bitrate**: 31 749 bps;
- **Vídeo bitrate e resolução**:

- 100 000 bps, resolução: 256 *x* 144;
- 264 835 bps, resolução: 426 *x* 240;
- 686 521 bps, resolução: 640 *x* 360;
- 869 460 bps, resolução: 854 *x* 480;
- 2 073 921 bps, resolução: 1280 *x* 720;
- 4 190 760 bps, resolução: 1920 *x* 1080.

A escolha deste *dataset* foi guiada por critérios relevantes para a avaliação do desempenho das abordagens de distribuição de carga em diferentes condições de rede e qualidade de serviço. Primeiramente, a variação nas qualidades de vídeo permite testar como cada abordagem gerencia a alocação de recursos sob diferentes demandas de largura de banda, refletindo cenários realistas onde a qualidade de transmissão pode variar conforme a capacidade da rede e a carga atual.

Além disso, a utilização de compressão x264, uma tecnologia amplamente utilizada em transmissões de vídeo devido à sua eficiência e qualidade, que de acordo com (Bitmovin, 2024) tem aproximadamente 84% de adoção nas plataformas atuais, assegura que os resultados obtidos sejam representativos de situações encontradas em ambientes reais de transmissão e distribuição de conteúdo.

Esses fatores combinados tornam o *dataset CAR CENC* uma escolha apropriada para avaliar a eficácia das diferentes abordagens de distribuição de carga, permitindo a análise do desempenho em uma variedade de condições operacionais.

5.1.5 Ferramentas Utilizadas para os Experimentos

Os experimentos utilizaram das seguintes ferramentas para execução:

- Sistema Operacional: Ubuntu 18.04.1 *Long Term Support* (LTS);
- Servidor HTTP Apache: versão 2.4.29 - necessário para disponibilização de conteúdos multimídia;
- Navegador Google Chrome: versão 125.0.6396.3;
- dash.js: Reprodutor MPEG-DASH na versão 4.7.4;
- O tamanho do *buffer* de vídeo do *dash.js* foi definido como o padrão de 20 segundos;
- Controlador Ryu: versão 4.34 - controlador da Rede Definida por Software;
- Open vSwitch: versão 2.9.8 - *switch* OpenFlow 1.3;

5.1.6 Cenário de Testes

Os testes distribuídos foram concebidos para avaliar a abordagem proposta em um ambiente representativo do mundo real, onde há distâncias consideráveis entre o usuário e o servidor. Esse teste foi novamente implementado com o auxílio da plataforma Cloudlab, que oferece múltiplos sítios para a implantação de máquinas virtuais. A Figura 9 ilustra o cenário utilizado para a realização dos testes, destacando as localizações geográficas dos servidores e a topologia da rede.

Os nós são classificados conforme a seguinte nomenclatura: nós que iniciam com a letra *S* representam *switches*, *U* denota nós de usuários, *E* refere-se a servidores de borda, *O* indica servidores de origem, e *C* corresponde a controladores. Essa nomenclatura facilita a identificação e o entendimento do papel de cada componente na rede. No total foram utilizados 6 servidores de borda, 8 usuários, 3 *switches*, 2 controladores e 1 servidor de origem.

Cada região disponibiliza um tipo específico de hardware contudo, a virtualização dos recursos é a mesma para todos os dispositivos virtualizados. Os detalhes dos hardwares são:

- *amd124* - AMD EPYC 7302P 16 núcleos 3.0GHz, 8GB RAM, SSD
- *sm110p-10s10615* - Intel Xeon Silver 4314 16 núcleos 2.4GHz, 8GB RAM, SSD
- *clnode145* - Intel E5-2683v3 14 núcleos 2.0GHz, 8GB RAM, SSD

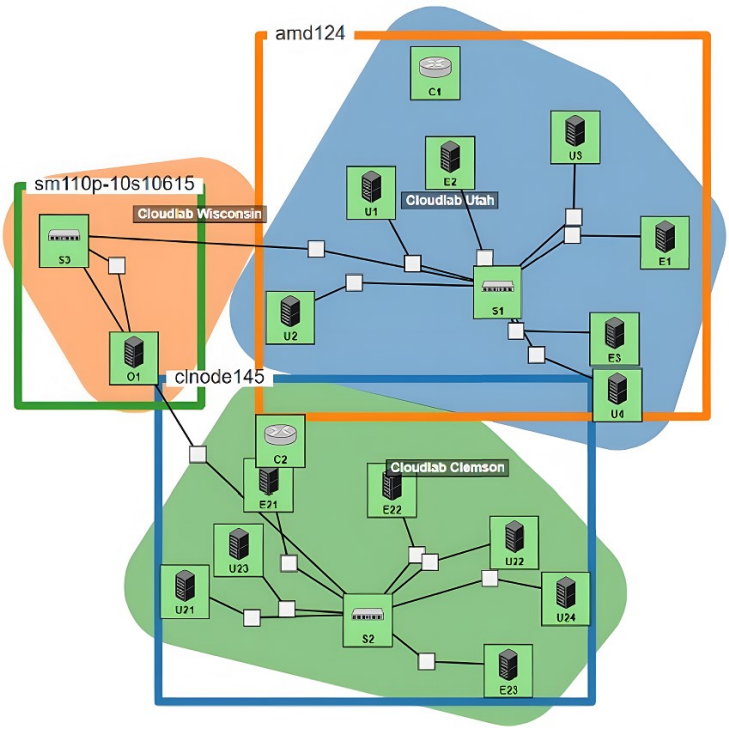
Todos os dispositivos virtualizados utilizados no experimento possuem as seguintes especificações:

- 1 GB de RAM;
- 2 núcleos de processamento e a velocidade depende da região conforme as especificações de hardware;
- Armazenamento em SSD;
- Largura de banda de 100Mb.

Embora essas especificações possam parecer modestas, elas são consideradas adequadas para os propósitos do experimento. Isso se deve ao fato de que o principal objetivo é avaliar a eficácia da abordagem proposta em um ambiente distribuído. Para isso, é suficiente que as máquinas virtuais possam executar as tarefas básicas de servir e consumir conteúdo multimídia. As especificações mínimas garantem que o desempenho seja avaliado de forma justa, sem a interferência de recursos de hardware excessivos que poderiam mascarar potenciais limitações da abordagem proposta uma vez que seria necessário mais tráfego para exaurir os recursos virtualizados.

Ainda, conforme a Figura 9, três regiões foram escolhidas para a realização dos testes: a região de Clemson, a região de Utah e a região de Wisconsin. A seleção dessas áreas se deve ao

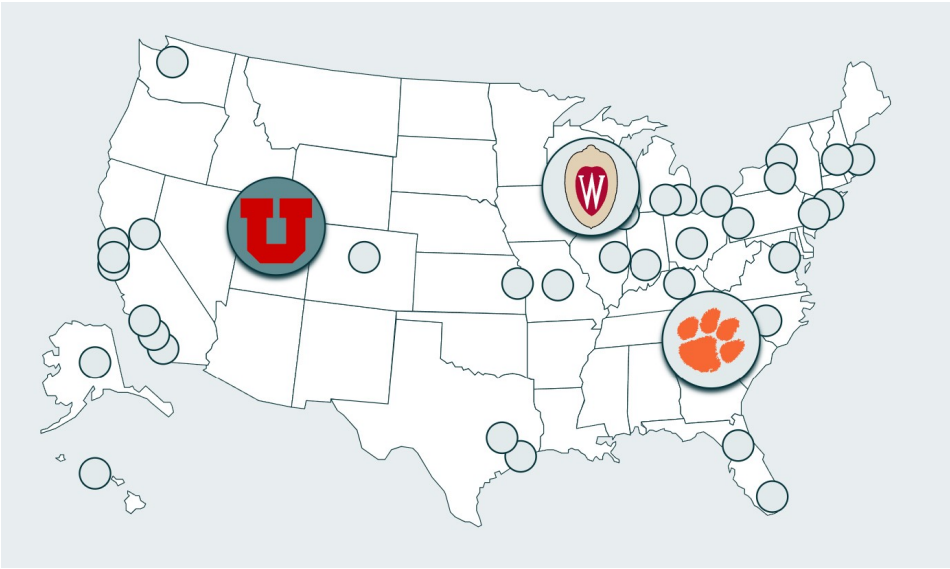
Figura 9 – Cenário Distribuído dos Experimentos



Fonte: O autor, 2024

fator geográfico e à dispersão significativa que elas apresentam entre si, conforme ilustrado na Figura 10. Essa distribuição geográfica permite uma avaliação mais rigorosa do desempenho da abordagem proposta em um ambiente de rede distribuída real. A ideia foi selecionar a região de Wisconsin como a região que hospede o servidor origem pois está mais centralizada e próxima das duas outras regiões.

Figura 10 – Mapa CloudLab



Fonte: cloudlab.us, 2024

Além disso, para análise comparativa, foram consideradas três abordagens distintas:

- **Sem redirecionamento:** Esta é a abordagem padrão, sem qualquer método de balanceamento de carga envolvido.
- **Estática:** Nesse caso, o redirecionamento do tráfego é avaliado apenas no início da reprodução do conteúdo, não ocorrendo periodicamente.
- **Dinâmica:** Esta é a abordagem proposta, com redirecionamento inicial conforme avaliação do controlador, além de redirecionamentos periódicos caso o controlador detecte outro servidor com métricas de desempenho superiores.

Devido a restrições de tempo, não foi possível utilizar o servidor de origem como um elemento ativo do cenário em todos os testes. Embora o servidor de origem esteja presente na configuração do sistema na região de Wisconsin, sua função original, que é repassar aos servidores de borda os conteúdos que eles não dispõem em seu catálogo, não foi executada. Em vez disso, o servidor de origem foi utilizado apenas como uma região adicional, o que aumentou a complexidade do sistema devido ao envolvimento de mais regiões. Essa limitação deve ser levada em consideração ao interpretar os resultados obtidos, pois sem o funcionamento completo do servidor de origem, a dinâmica real do sistema de distribuição de conteúdo não foi plenamente replicada. Essa questão também deverá ser abordada em experimentos futuros para garantir uma avaliação mais abrangente e representativa do desempenho do sistema sob condições reais de operação.

5.1.7 Experimentos e Resultados de Desempenho

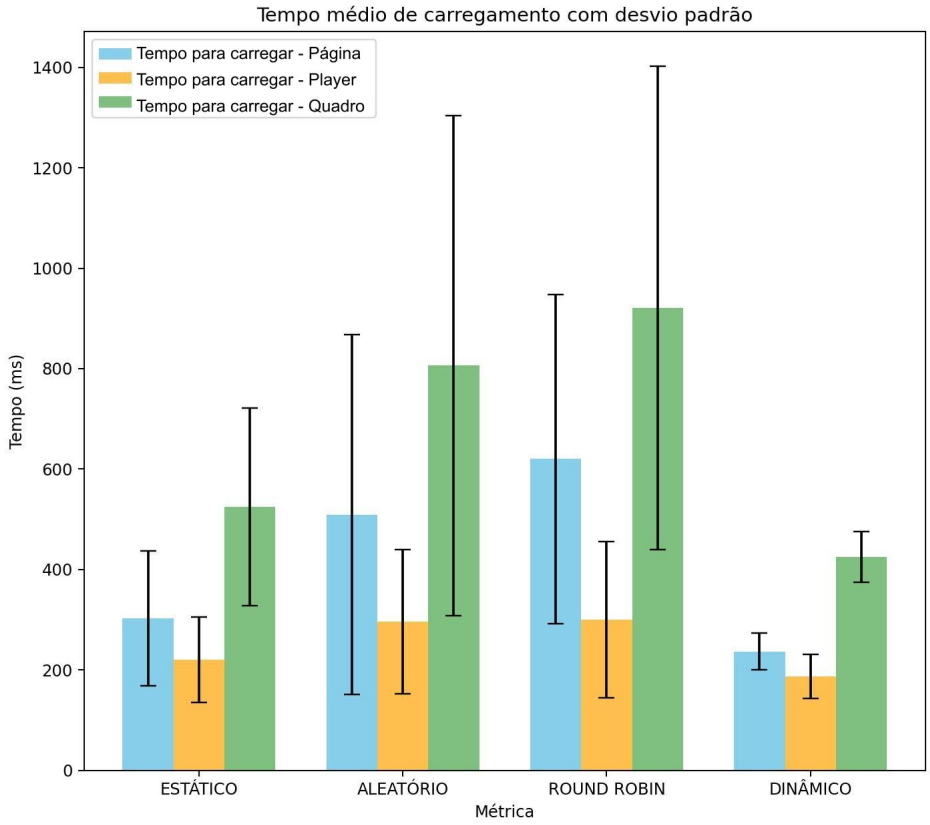
A Figura 11 apresenta os resultados referentes aos tempos de carregamento e ao aparecimento do primeiro quadro na transmissão de vídeo. Nesse experimento, foram testadas quatro abordagens, todas utilizando SDN e o controlador de diferentes formas.

No modo aleatório, o cliente é redirecionado para um servidor selecionado de maneira aleatória. No modo *round robin*, todos os servidores são acessados em uma ordem circular. As abordagens dinâmica e estática também foram incluídas na comparação.

Os resultados indicam que a abordagem utilizando o algoritmo de *round robin* apresentou os tempos mais longos de carregamento e de aparecimento do primeiro quadro. Em seguida, o método aleatório também demonstrou tempos elevados. Em contraste, as abordagens propostas, tanto no modo estático quanto no modo dinâmico, mostraram tempos significativamente menores em todas as categorias analisadas.

Adicionalmente, a abordagem dinâmica destacou-se por sua maior consistência nos tempos de resposta, evidenciada por um desvio padrão consideravelmente menor em comparação às outras abordagens. Isso sugere uma maior previsibilidade e eficiência no gerenciamento de recursos, o que é crucial para a otimização da experiência do usuário final em transmissões de vídeo.

Figura 11 – Tempo de carregamento (resposta) - Cenário distribuído com diversas abordagens de balanceamento de carga



Fonte: O autor, 2024

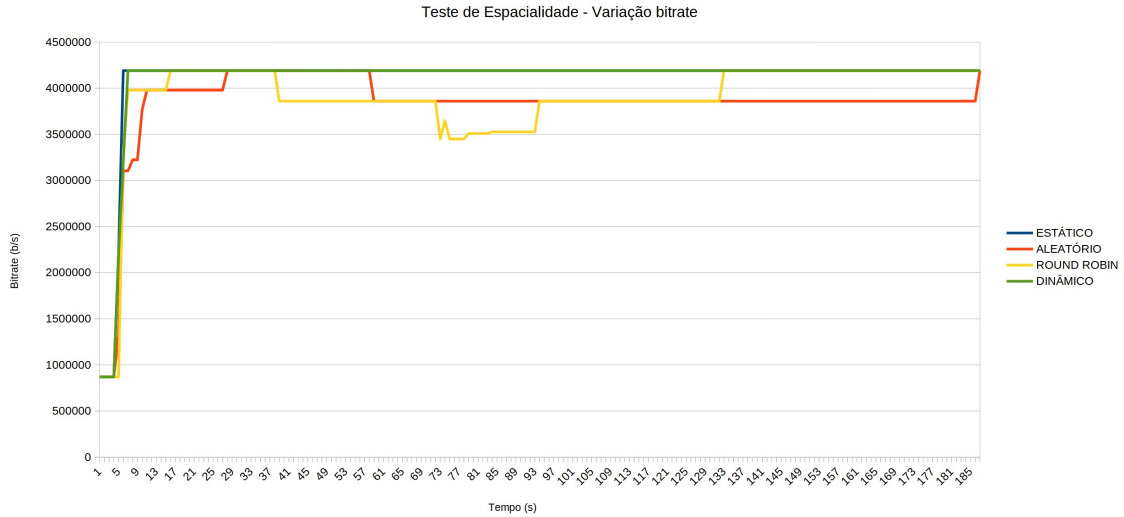
No teste de variação de *bitrate*, conforme a Figura 12, é possível perceber que tanto a abordagem dinâmica quanto a estática apresentaram maior consistência na qualidade durante todo o período de reprodução do vídeo. As abordagens *Round Robin* e aleatória, por outro lado, demonstraram diminuições na qualidade em alguns momentos, especialmente a abordagem de balanceamento *Round Robin*.

Essa diminuição de qualidade nas abordagens de *Round Robin* e aleatória ocorre porque elas redirecionam a conexão para servidores distantes. Devido ao provável congestionamento da rede, em determinadas situações, a qualidade do vídeo precisa ser reduzida para evitar interrupções (*stalls*) na reprodução.

A consistência no tempo de resposta observada nas abordagens dinâmica e estática pode ser atribuída à sua capacidade de gerenciar melhor os recursos da rede e de evitar servidores sobrecarregados ou muito distantes. Isso resulta em uma experiência de visualização mais suave e de alta qualidade, mesmo em condições de rede variáveis. Nenhuma das abordagens avaliadas apresentou *stall*.

Os testes de significância estatística utilizando a técnica de Kruskal-Wallis mostram que os tempos de carregamento apresentam o valor p de 0,029 que é menor que o nível de significância

Figura 12 – Variação do *Bitrate* - Cenário distribuído com diversas abordagens de balanceamento de carga



Fonte: O autor, 2024

comum de 0,05. Portanto, rejeita-se a hipótese nula, indicando evidências significativas de que existem diferenças entre as abordagens de balanceamento de carga testadas. A mesma técnica de avaliação estatística é utilizada para verificação da variação do *bitrate* que revela o valor p de $5,85 \times 10^{-75}$, muito inferior ao nível de significância comum de 0,05. Por isso, rejeita-se a hipótese nula, indicando evidências fortes de que existem diferenças significativas entre os grupos.

5.1.8 Experimentos e Resultados de Escalabilidade

Os testes de escalabilidade foram planejados para avaliar a eficácia da abordagem proposta em um ambiente caracterizado por uma demanda crescente de conexões. A metodologia adotada envolveu a simulação de um cenário onde, progressivamente, um número cada vez maior de conexões é recebido. Isso foi feito com o objetivo de verificar até que ponto a infraestrutura estabelecida consegue suportar a carga adicional sem comprometer o desempenho. À medida que o tempo avança, a quantidade de conexões aumenta, criando uma situação de sobrecarga que testa os limites da infraestrutura em termos de capacidade de processamento, estabilidade e resposta.

Esses testes são essenciais para identificar possíveis gargalos e pontos de falha que podem surgir sob condições de alta demanda. Além disso, fornecem dados que permitem otimizar a arquitetura, garantindo que a solução possa escalar de maneira eficiente e confiável. A análise dos resultados obtidos a partir desses testes contribui significativamente para a validação da robustez e da resiliência do sistema, bem como para o planejamento de futuras expansões e melhorias.

Devido ao tempo necessário para a execução completa de cada leva de testes, com dez execuções para cada nível de teste de escalabilidade proposto, apenas três abordagens foram

comparadas: a abordagem dinâmica, a aleatória e a de *Round Robin*.

Um *script* em Python foi desenvolvido para simular um ambiente com múltiplas conexões aos servidores presentes no ambiente de testes. O funcionamento do *script* ocorre da seguinte maneira:

- Todos os servidores são listados em uma estrutura de dados apropriada;
- A cada 5 segundos, o *script* seleciona um servidor da lista e estabelece uma determinada quantidade de conexões com esse servidor;
- Esse processo é contínuo, ocorrendo repetidamente ao longo de toda a execução do conteúdo multimídia.

O *script* foi projetado para testar a escalabilidade e a robustez dos servidores ao lidar com um aumento progressivo de conexões. Ao conectar-se periodicamente a diferentes servidores, o *script* simula um ambiente de uso realista, onde a carga sobre os servidores aumenta gradualmente.

5.1.8.1 Escalabilidade com 100 Conexões

O primeiro teste visa avaliar o ambiente sob condições mais simples. Nesse cenário, a cada 5 segundos, um servidor aleatório da rede recebe 100 conexões, com no máximo 10 conexões simultâneas. Essa configuração inicial serve como uma linha de base para comparar a eficácia das diferentes abordagens de distribuição de carga.

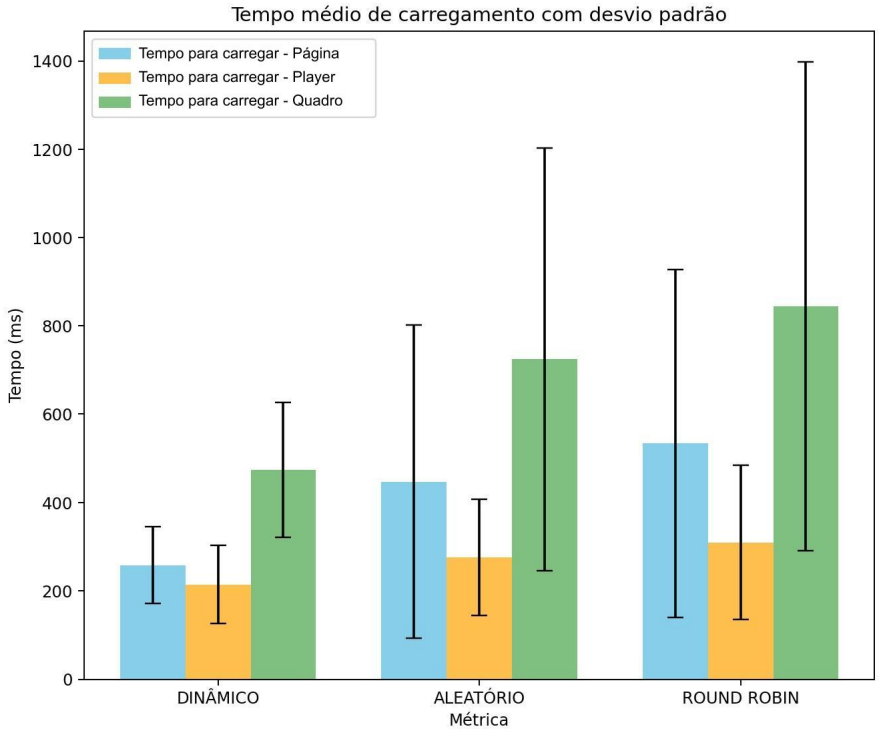
A abordagem dinâmica ajusta a distribuição de conexões com base no desempenho atual dos servidores, buscando otimizar a utilização dos recursos de forma adaptativa. A abordagem aleatória distribui as conexões sem seguir um padrão específico, proporcionando uma visão sobre o impacto da aleatoriedade na escalabilidade. Por sua vez, a abordagem *Round Robin* distribui as conexões de maneira equitativa e cíclica entre os servidores, garantindo que todos recebam uma quantidade semelhante de conexões ao longo do tempo. O seguinte comando é executado a cada 5 segundos em um servidor aleatório:

```
ab -n 100 -c 10 -k [servidor]
```

Conforme ilustrado na Figura 13, o tempo de carregamento da abordagem dinâmica mostrou-se consideravelmente mais ágil em comparação com as outras duas abordagens. Esse resultado evidencia a eficácia da seleção do servidor menos ocupado, característica principal da abordagem dinâmica.

O primeiro teste, portanto, revelou-se positivo, destacando os benefícios dessa abordagem. A capacidade de ajustar dinamicamente a distribuição das conexões com base no estado atual dos servidores permitiu uma utilização mais eficiente dos recursos disponíveis, resultando em tempos de resposta mais rápidos e melhor desempenho geral do sistema.

Figura 13 – Tempo de resposta - teste de escalabilidade 100 10



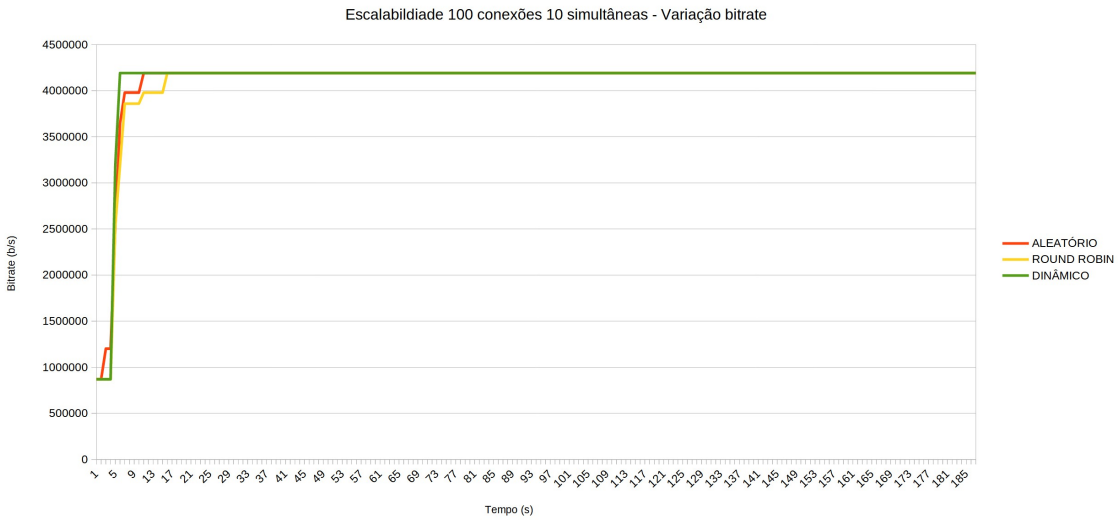
Fonte: O autor, 2024

A Figura 14 apresenta os resultados da variação do *bitrate* durante a reprodução do vídeo nesse cenário. Novamente, a abordagem dinâmica demonstrou ser mais consistente, proporcionando uma qualidade de experiência mais alta de forma mais rápida em comparação com as outras abordagens. Na abordagem dinâmica, a seleção do servidor menos ocupado permitiu uma entrega de dados mais eficiente, resultando em uma experiência de vídeo mais estável e de alta qualidade.

Por outro lado, a abordagem aleatória mostrou pontos de perda de qualidade, evidenciando sua falta de eficiência na distribuição de carga. A distribuição aleatória de conexões levou a uma utilização desigual dos servidores, resultando em variações no *bitrate* e, consequentemente, em uma experiência de reprodução de vídeo menos satisfatória. Nenhuma das abordagens avaliadas apresentou *stall*.

Os testes de significância estatística sugerem que o tempo de carregamento possui diferenças significativas pois utilizando a técnica de Kruskal-Wallis o valor de p é de 2.38×10^{-6} e portanto inferior ao valor comum denotado por 0.05, assumindo um intervalo de confiança de 95%. Já com relação a variação de *bitrate* o valor de p é 0.0812, logo não significativa entre as diferentes abordagens de balanceamento, nesse caso.

Figura 14 – Variação *bitrate* - teste de escalabilidade 100 10



Fonte: O autor, 2024

5.1.8.2 Escalabilidade com 1000 Conexões

O segundo teste de escalabilidade traz um aumento significativo no número de conexões. Nesse cenário, a cada 5 segundos, um servidor aleatório da rede recebe 1000 conexões, com no máximo 100 conexões sendo simultâneas. Este aumento substancial na carga tem o objetivo de avaliar como cada abordagem de balanceamento lida com condições de alta demanda, testando os limites da infraestrutura. Para obtenção dos resultados, o seguinte comando é executado a cada 5 segundos em um servidor aleatório:

```
ab -n 1000 -c 100 -k [servidor]
```

A Figura 15 ilustra os tempos de carregamento para todas as abordagens testadas no cenário de 1000 conexões. A abordagem dinâmica demonstrou novamente ser a mais consistente e rápida quando comparada às outras abordagens.

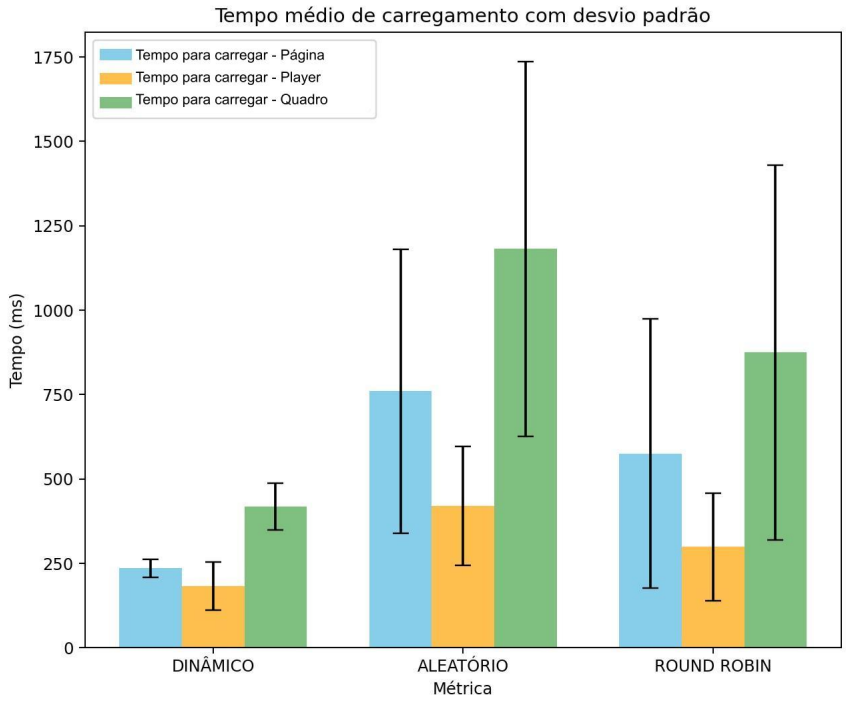
Os tempos de carregamento mais rápidos da abordagem dinâmica destacam sua eficácia em ajustar a alocação de conexões com base no estado atual dos servidores, permitindo uma melhor distribuição da carga e uma utilização mais eficiente dos recursos. Em contraste, as abordagens aleatória e *Round Robin* apresentaram variações maiores nos tempos de carregamento, indicando uma menor eficiência na gestão da carga em condições de alta demanda.

A Figura 16 mostra que não houve grande variação no *bitrate* ao longo da reprodução do vídeo para as diferentes abordagens testadas. Contudo, é possível notar uma leve vantagem da abordagem dinâmica, que consegue atingir a melhor qualidade mais rapidamente em comparação com as outras abordagens. Nenhuma das abordagens avaliadas apresentou *stall*.

Esses resultados reforçam a superioridade da abordagem dinâmica em ambientes de alto desempenho, onde a rapidez e a consistência no tempo de resposta são cruciais para a manutenção da qualidade do serviço e da experiência do usuário.

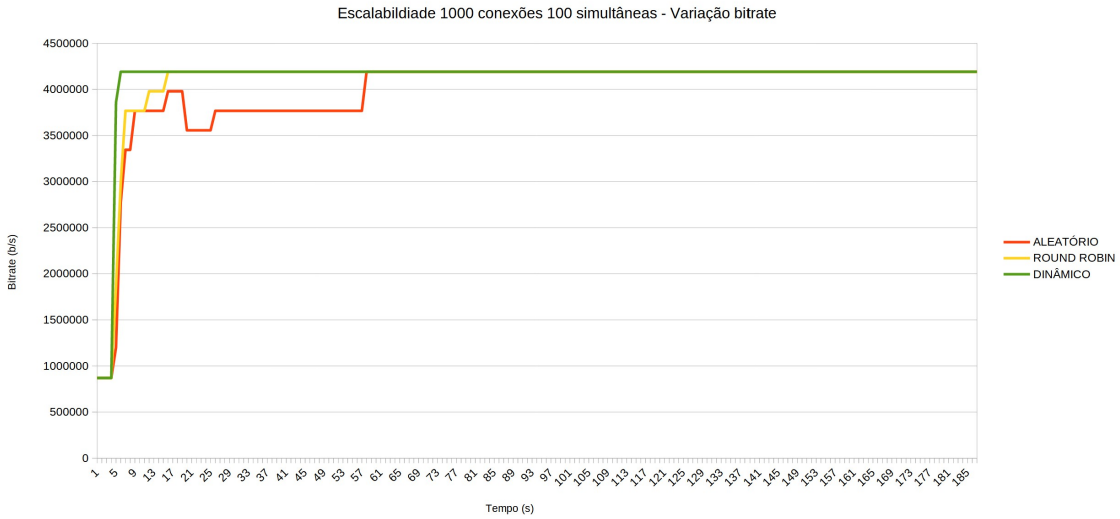
Os testes de significância estatística sugerem que o tempo de carregamento possui

Figura 15 – Tempo de carregamento (resposta) - teste de escalabilidade 1000 100



Fonte: O autor, 2024

Figura 16 – Variação *bitrate* - teste de escalabilidade 1000 100



Fonte: O autor, 2024

diferenças significativas pois utilizando a técnica de Kruskal-Wallis o valor de p é de 3.39×10^{-7} e portanto inferior ao valor comum denotado por 0.05. A variação de *bitrate* apresenta o valor de p igual a 6.11×10^{-11}, portanto também apresenta diferença significativa entre as diversas abordagens de balanceamento de carga.

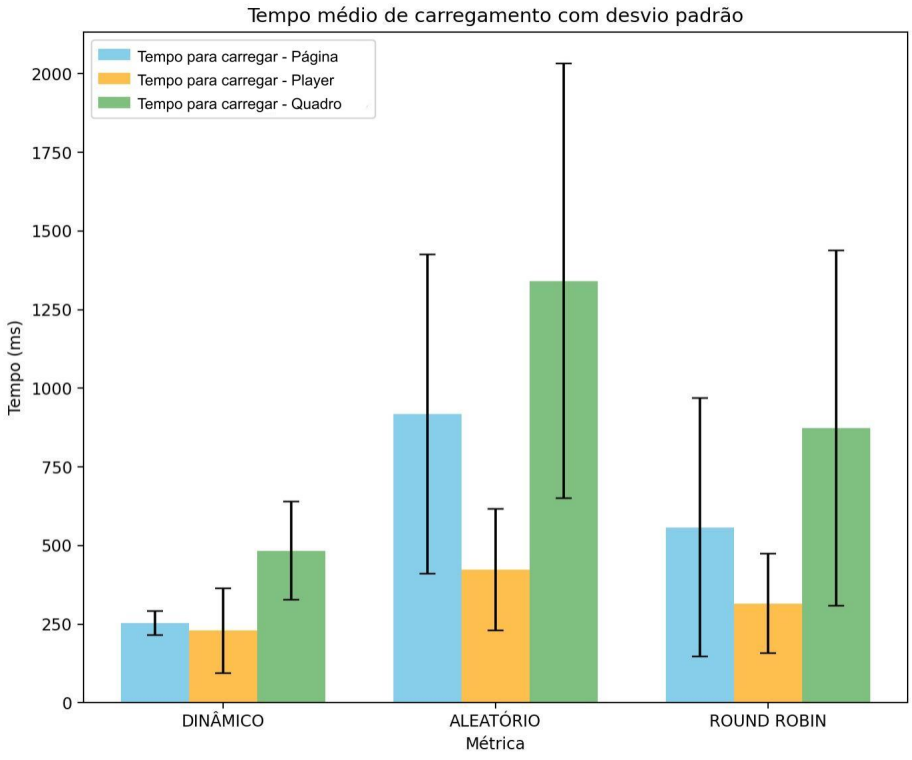
5.1.8.3 Escalabilidade com 5000 Conexões

O terceiro e último teste de escalabilidade visa saturar a rede de maneira geral a ponto de potencialmente afetar a qualidade da experiência do usuário. Nesse cenário, a cada 5 segundos, um servidor aleatório recebe até 5000 conexões, com no máximo 500 conexões sendo simultâneas. Esse teste foi idealizado para examinar como cada abordagem lida com condições extremas de alta demanda, forçando a infraestrutura ao seu limite. O seguinte comando é executado a cada 5 segundos em um servidor aleatório para obtenção dos resultados:

```
ab -n 5000 -c 500 -k [servidor]
```

A Figura 17 ilustra os tempos de carregamento, demonstrando novamente que a abordagem dinâmica se mantém mais eficiente que as outras abordagens. Em cenários de alta demanda, a capacidade da abordagem dinâmica de ajustar a distribuição de conexões em tempo real permite tempos de resposta mais rápidos e consistentes, evidenciando sua superioridade em condições de extrema carga.

Figura 17 – Tempo de resposta - teste de escalabilidade 5000 500

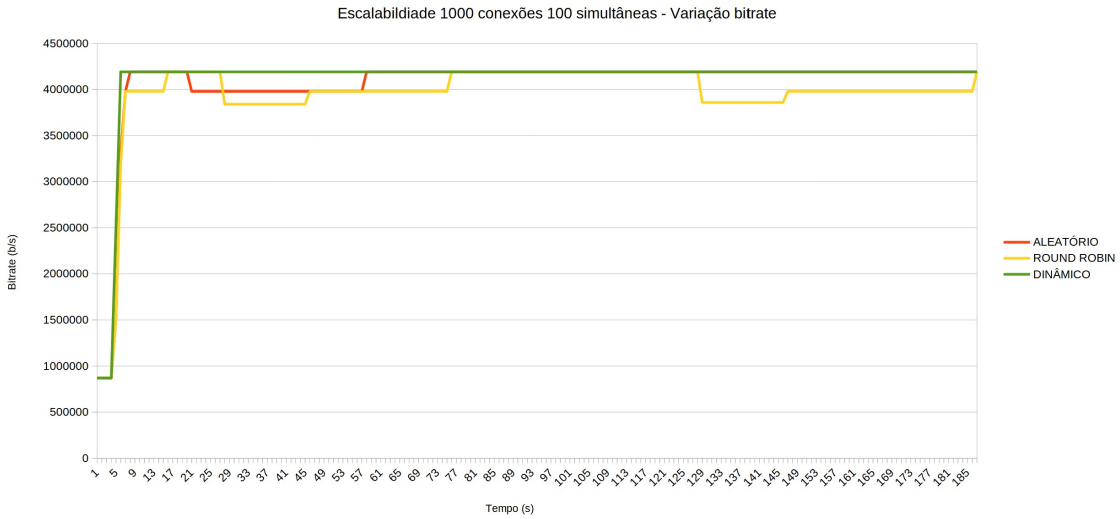


Fonte: O autor, 2024

Conforme mostrado na Figura 18, a variação de *bitrate* foi mais inconsistente neste cenário de teste, refletindo o impacto da alta saturação na rede. Contudo, a abordagem dinâmica conseguiu manter a qualidade máxima durante todo o período de testes. Essa consistência na qualidade do *bitrate* ressalta a eficiência da abordagem dinâmica em gerenciar a carga e otimizar

a entrega de dados, garantindo uma experiência de usuário superior mesmo sob condições de estresse extremo. Nenhuma das abordagens avaliadas apresentou *stall*.

Figura 18 – Variação *bitrate* - teste de escalabilidade 5000 500



Fonte: O autor, 2024

Esses resultados confirmam que, em situações de pico extremo, a abordagem dinâmica não só é mais eficiente em tempos de carregamento, mas também mantém a qualidade do serviço em níveis elevados, consolidando sua eficácia em ambientes de alta demanda. Ainda conforme discutido na Seção 3, o desafio de gerenciamento de conexões fica mais compreensível após a apresentação dos resultados obtidos uma vez que a abordagem de balanceamento de carga baseada no algoritmo de *Round Robin* não obteve resultados visivelmente inferiores ao da abordagem apresentada.

Os testes de significância estatística sugerem que o tempo de carregamento possui diferenças significativas pois utilizando a técnica de Kruskal-Wallis o valor de p é de 4.6×10^{-6} e portanto inferior ao valor comum denotado por 0.05, considerando um intervalo de confiança de 95%. Já com relação a variação de *bitrate* o valor de p foi de 8.12×10^{-38}, um valor extremamente baixo o que prova a diferença significativa.

5.2 RESUMO DO CAPÍTULO

Este capítulo apresenta a metodologia de pesquisa adotada, destacando as tecnologias utilizadas, a descrição dos experimentos realizados e a análise dos resultados obtidos. A escolha do CloudLab como plataforma de pesquisa foi fundamentada em sua flexibilidade, distribuição geográfica abrangente e capacidade de personalização. Os testes foram conduzidos em um cenário controlado, utilizando máquinas virtuais e a análise dos resultados concentrou-se em métricas relevantes para a qualidade de experiência do usuário, como a média do *bitrate* por segundo, o número de mudanças na qualidade do vídeo, o tempo de carregamento da página, o tempo para a aparição do primeiro quadro do vídeo no *player* e o número de *stalls* no

vídeo. A abordagem dinâmica de redirecionamento se destacou, evidenciando maior eficiência e consistência na entrega de conteúdo em comparação com as outras estratégias. Dessa forma a análise dos resultados revelou que a abordagem dinâmica proporciona tempos de carregamento mais rápidos, garantindo uma qualidade de experiência superior.

6 CONSIDERAÇÕES FINAIS

Com a proliferação da Internet e os avanços contínuos na arquitetura e estrutura das redes, o consumo diário de vídeos tornou-se uma prática ubíqua na rotina de muitos indivíduos. Entretanto, apesar dos significativos progressos e das melhorias constantes, ainda subsistem diversos desafios a serem enfrentados. Entre esses desafios, destaca-se a problemática do balanceamento de carga, no qual a rede de distribuição de conteúdo deve suportar uma demanda crescente de requisições sem comprometer a qualidade da experiência do usuário final.

Este estudo busca oferecer uma solução viável, ou pelo menos uma atenuação significativa, para o desafio de otimizar a transmissão de vídeo em redes de distribuição de conteúdo. Nesse contexto, propõe-se a implementação de um sistema baseado em Redes Definidas por Software (SDN) para interceptar pacotes durante a reprodução de conteúdo em vídeo. O sistema analisa as métricas de desempenho dos servidores de conteúdo disponíveis e, com base nessas análises, realiza uma seleção criteriosa do servidor mais apropriado para atender às requisições de conteúdo dos clientes.

Os resultados obtidos com essa abordagem são promissores. Em todas as situações em que o tempo foi a métrica avaliada, a abordagem se mostrou mais ágil e consistente, conforme evidenciado pelo baixo desvio padrão dos resultados. Além disso, nos testes focados na observação da flutuação do *bitrate*, houve uma notável consistência na reprodução do conteúdo. Uma vez que a qualidade máxima era atingida, essa se mantinha estável até o fim da reprodução do vídeo.

Adicionalmente, mesmo com o redirecionamento dinâmico das conexões durante a transmissão, não foram observadas alterações perceptíveis na qualidade do vídeo. Essa estabilidade é crucial para garantir uma experiência de usuário satisfatória, demonstrando que a técnica pode ser aplicada em cenários reais sem comprometer a qualidade da transmissão.

Em resumo, a aplicação de SDN para otimização de transmissão de vídeo mostra-se uma estratégia eficaz. A seleção dinâmica e inteligente dos servidores, baseada em métricas de desempenho, pode melhorar significativamente a qualidade da experiência do usuário e a eficiência da rede, tornando-se uma alternativa promissora para os desafios enfrentados na distribuição de conteúdo multimídia.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros, há várias possibilidades a serem exploradas para aprimorar e expandir a pesquisa atual. Primeiramente, executar testes utilizando vídeos de resoluções mais altas, como 4K e 8K, é uma outra direção importante. A transmissão de vídeos em alta resolução é cada vez mais comum, e esses testes poderão fornecer *insights* sobre o desempenho do sistema em condições mais exigentes. Será possível verificar a capacidade do sistema de manter a QoE dos usuários mesmo com o aumento na demanda de largura de banda e processamento que vídeos de alta resolução impõem.

A implementação e comparação da abordagem de *Content Steering* com a solução proposta é desejável. O *Content Steering* envolve a orientação dinâmica do conteúdo para melhorar a eficiência da entrega e a QoE. Comparar os resultados dessa abordagem com a solução atual permitirá identificar vantagens e desvantagens de cada técnica, além de possibilitar o desenvolvimento de soluções híbridas que possam combinar os pontos fortes de ambas. Explorar essas direções permitirá não apenas validar a robustez e a eficácia da solução atual em diferentes cenários e condições, mas também abrirá novas oportunidades para otimizar ainda mais a distribuição de conteúdo multimídia utilizando tecnologias emergentes e abordagens inovadoras.

Outra área a ser investigada é a exploração de técnicas de previsão de popularidade e o pre-posicionamento de conteúdo. A implementação dessas técnicas pode contribuir para a redução da latência e do consumo de largura de banda, otimizando ainda mais a experiência do usuário. A investigação nessa área abre oportunidades para explorar diferentes estratégias de *caching* e pré-posicionamento, considerando fatores como tipo de conteúdo, popularidade e padrões de acesso dos usuários. Isso pode resultar em um sistema mais eficiente e responsivo, capaz de fornecer conteúdo de maneira mais rápida e com menor consumo de recursos.

Por fim, é importante realizar um estudo aprofundado da percepção do usuário, além dos indicadores técnicos de desempenho, como latência e *throughput*. Avaliar o impacto da abordagem proposta na qualidade da experiência do usuário pode ser feito através de pesquisas de opinião, testes de usabilidade e outras metodologias que permitam capturar a percepção dos usuários sobre a fluidez, qualidade de imagem e outros aspectos relevantes da experiência de visualização de vídeo. Essa análise é crucial para garantir que as melhorias técnicas se traduzam em uma melhor experiência para os usuários finais.

REFERÊNCIAS

AFZAL, Shahbaz; GANESH, Kavitha. Load balancing in cloud computing -a hierarchical taxonomical classification. **Journal of Cloud Computing**, v. 8, 12 2019. Citado na página 23.

ANALYSTS, Inc Global Industry. **Cloud Computing Services - Global Market Trajectory & Analytics**. 2021. Citado na página 13.

ANDJAMBA, Taleni Shirley; ZODI, Guy-Alain Lusilao. A load balancing protocol for improved video on demand in sdn-based clouds. In: **2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM)**. [S.l.: s.n.], 2023. p. 1–6. Citado 2 vezes nas páginas 32 e 34.

BACCOUR, Emna et al. Proactive video chunks caching and processing for latency and cost minimization in edge networks. In: **2019 IEEE Wireless Communications and Networking Conference (WCNC)**. [S.l.: s.n.], 2019. p. 1–7. Citado na página 27.

BAMHDI, Alwi M. Cdca: Transparent cache architecture to improve content delivery by internet service providers. **International Journal of Advanced Computer Science and Applications**, The Science and Information Organization, v. 14, n. 10, 2023. Disponível em: <http://dx.doi.org/10.14569/IJACSA.2023.0141090>. Citado 2 vezes nas páginas 32 e 34.

Bitmovin. **The 7th Annual Bitmovin Video Developer Report**. [S.l.], 2024. Citado 2 vezes nas páginas 19 e 56.

BOURKE, Tony. **Server load balancing**. 1st. ed. [S.l.]: O'Reilly, 2001. ISBN 978-0-596-00050-9. Citado na página 26.

BUKHARI, Syed M. A. H.; AFAQ, Muhammad; SONG, Wang-Cheol. Streaming via sdn: Resource forecasting for video streaming in a software-defined network. In: **2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)**. [S.l.: s.n.], 2023. p. 596–601. Citado 2 vezes nas páginas 33 e 34.

CHIANG, Wei-Kuo; LI, Tsung-Ying. An extended sdn architecture for video-on-demand caching. **Mobile Networks and Applications**, p. 1–18, 04 2024. Citado 2 vezes nas páginas 33 e 34.

CLOUDLAB. **The CloudLab Manual**. 2023. Acesso em: 10 out. 2023. Disponível em: <https://docs.cloudlab.us/>. Citado na página 54.

COMER, Douglas E. **Redes de computadores e internet**. 6th. ed. Bookman Companhia Editora Ltda, 2016. ISBN 9788582603734. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788582603734/>. Citado na página 17.

CUCCHIARA, R.; PICCARDI, M.; PRATI, A. Neighbor cache prefetching for multimedia image and video processing. **IEEE Transactions on Multimedia**, v. 6, n. 4, p. 539–552, 2004. Citado na página 30.

DASH.JS. **A reference client implementation for the playback of MPEG DASH via Javascript and compliant browsers.: Dash-Industry-Forum/dash.js**. [S.l.]: Dash Industry Forum, 2012. Disponível em <https://github.com/Dash-Industry-Forum/dash.js>. Acesso 13 Out. de 2018. Citado na página 55.

DERNBACH, Stefan et al. Cache content-selection policies for streaming video services. In: **IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications**. [S.l.: s.n.], 2016. p. 1–9. Citado na página 29.

FISHER, R.A. **Statistical methods for research workers**. 1st. ed. [S.l.]: Edinburgh Oliver & Boyd, 1925. 262 p. ISBN 978-8-1307-0133-2. Citado na página 47.

FOUNDATION, The Apache Software. **ab - Apache HTTP server benchmarking tool**. 2024. Disponível em: <https://httpd.apache.org/docs/2.4/programs/ab.html>. Citado na página 44.

GORANSSON, Paul; BLACK, Chuck. **Software Defined Networks: A Comprehensive Approach**. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2014. ISBN 012416675X, 9780124166752. Citado na página 18.

GROUP, DASH-IF Interoperability Working. **Content Steering for DASH**. [S.l.]: DASH-IF Interoperability Working Group, 2022. Disponível em <https://dashif.org/docs/DASH-IF-CTS-00XX-Content-Steering-Community-Review.pdf>. Acesso 8 Ago. de 2024. Citado 2 vezes nas páginas 32 e 34.

HAMADAH, Siham. A survey: A comprehensive study of static, dynamic and hybrid load balancing algorithms. **International Journal of Computer Science, Information Technology and Security (IJCSITS)**, v. 7, p. 2249–9555, 03 2017. Citado na página 22.

HARIS, Mohammad; KHAN, Rafiqul Zaman. A systematic review on load balancing tools and techniques in cloud computing. In: SUMA, V. et al. (Ed.). **Inventive Systems and Control**. Singapore: Springer Nature Singapore, 2022. p. 503–521. ISBN 978-981-19-1012-8. Citado na página 13.

HASSLINGER, Gerhard et al. Optimum caching versus lru and lfu: Comparison and combined limited look-ahead strategies. In: **2018 16th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)**. [S.l.: s.n.], 2018. p. 1–6. Citado na página 30.

ISO. **ISO/IEC 23009-1:2014 - Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats**. 2014. Disponível em <https://www.iso.org/standard/65274.html>. Acesso 06 Set. de 2023. Citado na página 20.

KARAMCHANDANI, Nikhil et al. Hierarchical coded caching. **IEEE Transactions on Information Theory**, v. 62, n. 6, p. 3212–3229, 2016. Citado na página 28.

KILLELEA, P. **Web Performance Tuning: Speeding Up the Web**. O'Reilly Media, Incorporated, 2002. (O'Reilly Series). ISBN 9780596001728. Disponível em: <https://books.google.com.br/books?id=sX60mAi0eQUC>. Citado na página 22.

KIM, Hyong-youb; RIXNER, Scott. Tcp offload through connection handoff. In: **Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006**. New York, NY, USA: Association for Computing Machinery, 2006. (EuroSys '06), p. 279–290. ISBN 1595933220. Disponível em: <https://doi.org/10.1145/1217935.1217962>. Citado na página 39.

KLöTI, R.; KOTRONIS, V.; SMITH, P. Openflow: A security analysis. In: **2013 21st IEEE International Conference on Network Protocols (ICNP)**. [S.l.: s.n.], 2013. p. 1–6. ISSN 1092-1648. Citado na página 19.

KRUSKAL, William H.; WALLIS, W. Allen. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, Taylor & Francis, v. 47, n. 260, p. 583–621, 1952. Citado na página 46.

LAKATOS, Eva M. **Fundamentos de Metodologia Científica**. Grupo GEN, 2021. Acesso em: 30 ago. 2023. ISBN 9788597026580. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788597026580/>. Citado na página 15.

LIU, Dong; WANG, Zhiyong; ZHANG, Jie. Video stream distribution scheme based on edge computing network and user interest content model. **IEEE Access**, v. 8, p. 30734–30744, 2020. Citado 2 vezes nas páginas 31 e 34.

LUO, Yihui; CHANGSHENG, Xie; CHENGFENG, Zhang. Weighted cache replace algorithm for storage system. **International Journal of Computational Intelligence Systems**, 10 2007. Citado na página 29.

MAJDABADI, Reza Hedayati; WANG, Mea; RAKAI, Logan. Soda-stream: Sdn optimization for enhancing qoe in dash streaming. In: **NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium**. [S.l.: s.n.], 2022. p. 1–5. Citado 2 vezes nas páginas 31 e 34.

McKeown, Nick et al. OpenFlow: enabling innovation in campus networks. **ACM SIGCOMM Computer Communication Review**, v. 38, n. 2, p. 69, 2008. ISSN 01464833. Citado 2 vezes nas páginas 17 e 19.

MONTEIRO, Eduarda R et al. **Sistemas Distribuídos**. 1st. ed. SAGAH, 2020. ISBN 9786556901978. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9786556901978/>. Citado na página 28.

OLANREWAJU, Rashidah F. et al. A study on performance evaluation of conventional cache replacement algorithms: A review. In: **2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)**. [S.l.: s.n.], 2016. p. 550–556. Citado na página 30.

OMER, Yassin Abdulkarim Hamdalla; MOHAMMEDEL-AMIN, Mohamed Ayman; MUSTAFA, Amin Babiker A. Load balance in cloud computing using software defined networking. In: **2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)**. [S.l.: s.n.], 2021. p. 1–6. Citado na página 14.

ONF. **SDN architecture**. [S.l.]: ONF, 2014. Disponível em <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf>. Acesso 13 Ago. de 2018. Citado na página 18.

ONF. **OpenFlow Switch Specification v.1.5.1**. 2015. Disponível em <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>. Acesso 13 Ago. 2018. Citado na página 18.

Oracle. **Oracle Database Performance Tuning Guide**. [S.l.], 2003. Citado na página 25.

Oracle. **Oracle Sun Java System Application Server Enterprise Edition 8.2 Deployment Planning Guide**. [S.l.], 2007. Citado na página 25.

OVS. **Open vSwitch Release 2.10.90**. 2018. Disponível em <https://media.readthedocs.org/pdf/openvswitch/latest/openvswitch.pdf>. Acesso 30 Set. 2018. Citado na página 55.

PERRIN, Sterling; HUBBARD, Stan. **White Paper: Practical Implementation of SDN & NFV in the WAN**. [S.l.], 2013. 11 p. Citado na página 19.

SANDVINE. **Global Internet Phenomena**. 2023. Disponível em: <https://www.sandvine.com/phenomena>. Citado na página 13.

SANI, Yusuf; MAUTHE, Andreas; EDWARDS, Christopher. Adaptive bitrate selection: A survey. **IEEE Communications Surveys & Tutorials**, PP, p. 1–1, 07 2017. Citado 2 vezes nas páginas 19 e 20.

SHI, Weisong et al. Edge computing: Vision and challenges. **IEEE Internet of Things Journal**, v. 3, p. 1–1, 10 2016. Citado na página 27.

SILHAVY, Daniel et al. Latest advances in the development of the open-source player dash.js. In: **Proceedings of the 1st Mile-High Video Conference**. Denver Colorado: ACM, 2022. p. 32–38. ISBN 978-1-4503-9222-8. Citado na página 55.

STOLL, Julia. **OTT video revenue worldwide from 2010 to 2026**. 2021. Citado na página 13.

TAHA, Miran. An efficient software defined network controller based routing adaptation for enhancing qoe of multimedia streaming service. **Multimedia Tools and Applications**, v. 82, p. 1–24, 03 2023. Citado 2 vezes nas páginas 32 e 34.

TANENBAUM, A.S.; WETHERALL, D.J. **Redes de computadores**. 5th. ed. [S.l.]: Pearson Prentice Hall, 2011. ISBN 9788576059240. Citado na página 17.

TANENBAUM, Andrew S.; STEEN, Maarten Van. **Distributed Systems: Principles and Paradigms**. 1st. ed. USA: Prentice Hall PTR, 2001. ISBN 0130888931. Citado na página 39.

UBUNTU, Canonical Ltd. **Stress Tool**. 2019. Disponível em: <https://manpages.ubuntu.com/manpages/focal/man1/stress.1.html>. Citado na página 43.

WANG, Weikun; CASALE, Giuliano. Evaluating weighted round robin load balancing for cloud web services. In: **2014 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing**. [S.l.: s.n.], 2014. p. 393–400. Citado na página 23.

WANG, Zheng; HUANG, Jun; ROSE, Scott. Evolution and challenges of dns-based cdns. **Digital Communications and Networks**, v. 4, n. 4, p. 235–243, 2018. ISSN 2352-8648. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2352864817300731>. Citado na página 14.

WAZLAWICK, Raul S. **Metodologia de Pesquisa para Ciência da Computação**. Grupo GEN, 2020. Acesso em: 30 ago. 2023. ISBN 9788595157712. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788595157712/>. Citado na página 15.

WICHTLHUBER, Matthias; REINECKE, Robert; HAUSHEER, David. An sdn-based cdn/isp collaboration architecture for managing high-volume flows. **IEEE Transactions on Network and Service Management**, v. 12, p. 1–1, 03 2015. Citado 2 vezes nas páginas 24 e 25.

YOSHIHISA, Tomoki. A video pre-caching scheme based on power consumption on edge computing environments. In: **2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)**. [S.l.: s.n.], 2021. p. 445–447. Citado na página 27.

YOUTUBE. **MPEG-DASH / Media Source demo**. 2012. Disponível em: <http://yt-dash-mse-test.commondatastorage.googleapis.com/>. Citado na página 55.

APÊNDICE A – DISPONIBILIZAÇÃO DOS MATERIAIS UTILIZADOS

As aplicações utilizadas no trabalho, bem como os relatórios gerados pelo navegador contendo as estatísticas de cada reprodução, as modificações no player *dash.js*, os arquivos que contemplam a aplicação de redirecionamento dinâmico do controlador Ryu e o arquivo que contém o monitoramento de métricas referente ao modo dinâmico, estão disponíveis no link fornecido. Este repositório contém os componentes de código essenciais para replicar o ambiente de testes e validar os resultados apresentados, proporcionando transparência e facilitando futuras pesquisas e desenvolvimentos na área.

- <https://tinyurl.com/ppgcap-udesc-edenilson>

ÍNDICE REMISSIVO