Load Balancing to Optimize MPEG-DASH Video Streaming

Edenilson Jônatas dos Passos
Graduate Program in Apllied Computing (PPGCAP)
Department of Computer Science (DCC)
Santa Catarina State University (UDESC)
Joinville, Brazil
edenilson.passos@yahoo.com

Adriano Fiorese
Graduate Program in Apllied Computing (PPGCAP)
Department of Computer Science (DCC)
Santa Catarina State University (UDESC)
Joinville, Brazil
adriano.fiorese@udesc.br

***Abstract*—Recent advancements in on-demand multimedia streaming have revealed the potential of this business model, driven by the convenience of accessing content anytime and anywhere. However, delivering a seamless user experience remains challenging due to the demanding nature of audiovisual streams, despite significant resource allocation. A promising solution involves using load balancers to distribute workloads evenly among processing units. This study proposes optimizing the processing and transmission resources by means of Software-Defined Networks (SDN). By continuously monitoring performance metrics like CPU usage and throughput, the SDN network controller determines the best server for handling new connections, even allowing seamless server migration during playback. The results show up to 80% faster response times and more consistent video quality, with reduced latency and uninterrupted playback. The approach also confirms the feasibility of SDN in streaming services, providing a foundation for future network improvements.**

***Index Terms*—Load balancing, SDN, Monitoring, Video-streaming**

I. INTRODUCTION

The cloud computing and subscription-based video streaming markets have been consistently growing in recent years. With the advent of the COVID-19 pandemic, this growing have become even more significant. According to Global Industry Analysts [1], cloud computing services generated approximately $313.1 billion in revenue in 2020, and it is estimated that by 2027 this figure could reach around $947 billion.

Similarly, the video streaming services sector has experienced growth patterns comparable to those of cloud computing services. According to [2], this market segment achieved approximately $100 billion in revenue in 2020 and it is expected to surpass $200 billion by 2026. Additionally, according to [3], video-related content accounted for 65.93% of the total traffic volume on the Internet.

Given such growth, the space for new technologies and approaches to data transmission network infrastructure is promising, as service providers struggle to cope with increasing demand while maintaining adequate service quality. In this context, methods to mitigate this issue exist. One such method is load balancing. According to [4], the optimal choice of provider (i.e., computer, server, or service) made by the load balancer can benefit the system as a whole in various aspects, such as reducing the risk of failure and overload, improving scalability, reducing response times overall, and thereby enhancing customer satisfaction and significantly reducing system maintenance costs.

Load balancing can be addressed by means of various approaches. Performance metrics-based is one of which. Generally, when a request is made, the load balancing model selects the server best suited to attend the demand at that moment. This server selection process can be based on one or more predefined metrics or the overall network behavior.

In this work, an approach is presented for the equitable distribution of processing load by redirecting video content request and response traffic, focusing on monitoring and subsequently recovering the constituent metrics of the load balancing indicator through the adoption of the Software-Defined Networking (SDN) paradigm. This enables a metric-based load balancing method that operates directly at the session layer of the network infrastructure. Instead of relying on a specific server for load balancing, the equitable allocation of customer service load is performed by the packet-switching devices themselves, compatible with the OpenFlow (SDN) architecture, located in the network segment where the several servers organized in clusters are located. To this end, client request traffic is redirected to content servers that align with the load balancing policy.

This paper is organized as follows. Section II, discusses related works. Next, Section III details the proposed solution approach. Section IV presents the evaluation of the proposal, and Section V provides the final considerations of this work.

II. RELATED WORK

The work by [5] suggests a centralized control architecture, called Named Data Networking (NDN), which uses edge nodes to enhance Quality of Experience (QoE) and save bandwidth through caching and distributed processing for video-streaming applications. Similarly, [6] presents an optimization framework based on Dynamic Adaptive Streaming (DASH) to maximize the number of simultaneous sessions

979-8-3315-0694-0/25/$31.00 ©2025 IEEE

and streaming quality, utilizing SDN for dynamic routing and bandwidth allocation. Another study by [7] proposes the Content Steering technique to optimize video delivery, enabling dynamic routes between different Content Delivery Networks (CDNs) through a Media Presentation Description (MPD) file, a key component of DASH. The research by [8] introduces the Load Balancing Routing Protocol (LBRP), which adapts routing to improve user experience in video streaming by prioritizing livestream traffic. Meanwhile, [9] develops an artificial intelligence-based algorithm to optimize path selection in SDN networks, minimizing decision time and reducing resource usage on congested paths. The proposal by [10] combines Information Centric Networking (ICN) and SDN in a hybrid architecture to create a transparent caching system, improving Video on Demand (VoD) performance in traditional Internet Protocol (IP) networks. Additionally, [11] presents a method for predicting resources in SDN switches by applying machine learning algorithms in a video transmission scenario, while [12] proposes the Extended SDN Cache (ESC), an architecture that disaggregates caching functions to reduce the load on SDN controllers and increase the system's capacity and flexibility. Most previous studies have primarily focused on traffic balancing without adequately addressing load balancing, particularly in the context of video content distribution. Although a significant number of works utilize SDN for load balancing, few specifically focus on video distribution. The central innovation of this work lies in the ability of the proposed approach to manage high-volume and long-duration connections in a highly adaptive manner. This is achieved through the ability to manipulate connections in real-time, even during multimedia playback, allowing for load balancing optimization without disrupting the user experience.

III. Proposed Solution

To address the challenge of workload overload on multimedia content servers, this strategy employs continuous resource monitoring. The goal is to redirect traffic to the most suitable server for new connections without disrupting ongoing content playback. Although traffic redirection during playback can introduce issues like errors and delays, this research presents a seamless approach that operates unnoticed by users. This distinct feature highlights the contribution of this work, offering a more refined solution than those typically found in the literature.

Central to this approach is dynamic load balancing, driven by real-time server performance data. By efficiently redirecting traffic, the system optimizes resources and minimizes any impact on service quality. Moreover, Software-Defined Networking (SDN) plays a pivotal role, providing flexibility and control over network traffic. The integration of SDN facilitates real-time adjustments based on server conditions, ensuring a responsive system. Key metrics such as throughput, CPU and memory usage, and storage utilization are closely monitored due to their direct influence on video streaming performance. By continuously tracking these metrics and responding swiftly to any signs of overload, the system enhances performance, resilience, and service stability.

The Transmission Control Protocol (TCP) Handoff technique is already well-known in the field of communication networks, as it plays a crucial role in the efficient management of connections in distributed and highly dynamic environments. However, when applied to legacy infrastructure, its implementation proves to be highly complex, requiring specific hardware resources and technologies that are generally not part of the already established network infrastructure. Nevertheless, with the rise of Software-Defined Networking (SDN), this technique can be implemented in a less complex and even more efficient manner.

To migrate an active TCP session from a client, the controller executes the procedure of deleting the two corresponding flows, blocking communication between the old server and the client. As a result of this deletion, the next packet sent by the client is redirected to the SDN controller. New flows are then established between the client and the new server by the controller. It's important to note that the new server will not recognize the client's session, leading it to send a Reset (RST) packet in response, forcing the client to restart the TCP session through a Synchronize (SYN) packet, to which the new server responds with a SYN+ACK packet. From this point on, the client is able to request the necessary video segments from the new server. Next, the controller can forge two new packets based on the sequence and acknowledgment information extracted from the client's packet. These packets simulate the client's intention to close the session: FIN and ACK. The first informs the old server of the client's intention to end the session, and the server responds with a FIN+ACK packet. The controller can generate these two packets even without the old source's response, sending the first packet and waiting for a short interval of two milliseconds before sending the subsequent ACK packet, thereby appropriately closing the old TCP session. This process demonstrates the controller's ability to manage and orchestrate the transition of TCP sessions with precision and efficiency.

Moreover, by continuously monitoring resource usage on servers, it is possible to ensure that they are operating within or close to their capacity limits. The choice of the most suitable server to handle a connection is based on a simple ranking system in which each metric receives a score based on how detrimental its poor values can be to the final QoE performance.

In order to develop an effective indicator to identify the most suitable content server to attend a player's request, an stress test on CloudLab platform [13] was performed. This stress test took into account several MPEG-DASH content server performance metrics that are intrinsically related to measuring the client's QoE, covering various aspects that directly impact the user's perception and satisfaction. Therefore, average bitrate per second, number of quality changes in the video, time required to load the video web page, time taken to display the first video frame in the player and the number of video stalls. Considering that servers are using CPU, RAM memory, Disk (storage) and network throughput to deliver the video, it was possible to correlate them with the results of the video QoE stress test. As a result, it was found that disk access most influences the performance, followed by throughput and CPU and RAM usage by last ones even.

Thus, taking into account this found, the initial guidelines for redirecting connections to select the most appropriate server is proposed as follows. The server with the highest throughput is assigned 2 points. The server with the lowest CPU utilization receives 1 point, while the one with the lowest RAM utilization is also given 1 point. The server with the lowest disk utilization is awarded 3 points.

In this sense, define the scoring function Score(s) for a given server s as follows:

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

Where:

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

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

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

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

The controller plays a central role in the system by continuously monitoring server performance metrics and extracting critical data that feeds into an efficiency indicator. This is made possible through the integration of tools like the mod_status module and the Paramiko library, which enable precise, real-time monitoring. Mod_status provides detailed server status insights, including active processes, resource consumption, and workload, while Paramiko ensures secure SSH communication for remote metric collection. The controller not only monitors server states but also actively interacts with them for maintenance, configuration adjustments, and load redistribution. This combination allows for automated and responsive management, maintaining service quality and operational efficiency by identifying issues early and taking preventive actions.

A. System Load Balancing Architecture

The proposed architecture for the load balancing system is hierarchical, consisting of three levels. This approach enhances the system's robustness by grouping resources into hierarchical structures, which facilitates workload management and distribution, even as the system grows and resources spread across different geographic locations. Figure 1 illustrates the three proposed hierarchical levels.

The first layer, called the *Origin* includes the video-content origin servers. These servers play a critical role in a streaming service architecture as central repositories that store the entire media catalog available to users. They are characterized by significant computational capacity, enabling them to store and manage a large volume of content such as movies, series, music, and other types of media. However, the distinctive feature of origin servers is their physical location, typically in data centers or strategic locations. This geographical distance can be considerable, sometimes spanning entire regions or even countries. For efficiency and scalability reasons, connections are redirected to an origin node only when none of the edge servers have the requested content in their local repository.

Controller nodes are crucial as they receive the initial user requests and act as the *brain* of the system. These nodes house Software-Defined Networking (SDN) controllers, responsible for determining which servers are most suitable for handling the user's connection. When a request is received, the controller selects the most appropriate server. Once identified, the controller redirects the connection to that specific server, ensuring that user connections are optimized and directed to the most suitable servers, thereby enhancing overall system performance.

Edge servers, or server nodes, are strategically located near end-users to optimize performance and communication latency. However, these edge computing nodes have limited computational resources compared to origin servers. When a user makes a request, the controller identifies the most suitable server by constantly monitoring the server group to find nodes containing the requested content. After identifying potential candidates, the controller carefully evaluates them based on throughput, RAM, CPU, and disk usage to determine the best node to receive the user's connection.

IV. Experiments and Results

Scalability tests were designed to evaluate the effectiveness of the proposed approach in an environment characterized by increasing connection demand. The methodology involved simulating a scenario where a progressively larger number of connections is received by origin servers. The goal was to determine how well the established infrastructure can handle the additional load without compromising performance. As time progresses, the number of connections increases, creating an overload situation that tests the infrastructure's limits in terms of processing capacity, stability, and response time.

Tests were performed using the CloudLab platform [13], which is deployed over the whole USA country. Three geographical regions were chosen for the tests: Clemson, Utah, and Wisconsin depicted on Figure 2. The selection of these areas was based on their geographic dispersion because it allows for a more rigorous assessment of the proposed approach's performance in a real distributed network environment. The region of Wisconsin was selected to host the origin server since it is more centrally located and closer to the other two regions.

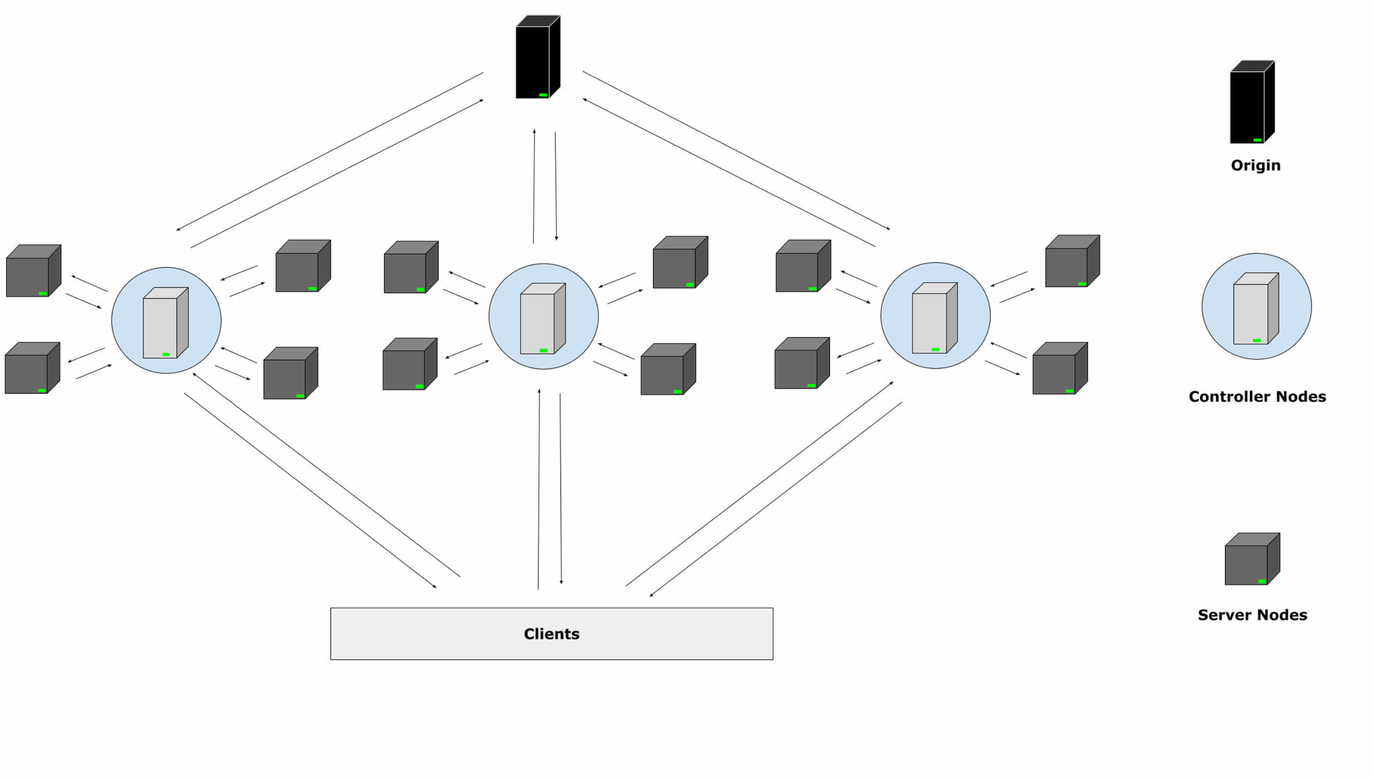

Fig. 1. Load Balancing System Architecture

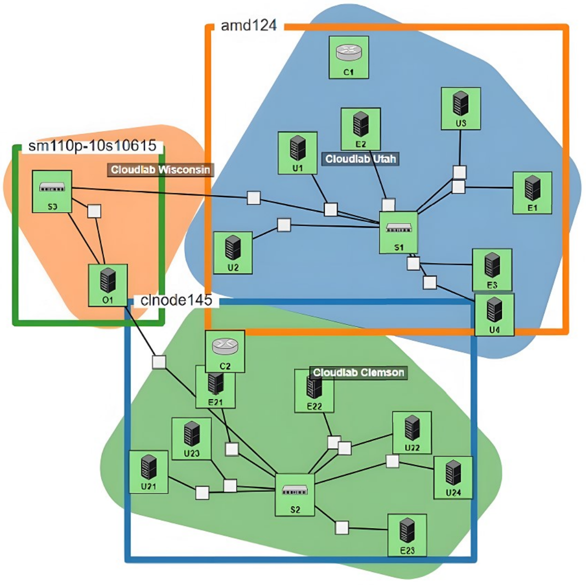

Fig. 2. Distributed Experiment Scenario

The first performed test aims to evaluate the environment under simpler conditions. In this scenario, every 5 seconds, a random server in the network receives 100 connections, with a maximum of 10 simultaneous connections. This initial configuration serves as a baseline to compare the effectiveness of different load distribution approaches.

Three different server's connection distribution approaches were tested coping with the load balancing system's scalability. The dynamic approach adjusts the distribution of connections based on the current performance of the servers, seeking to optimize resource utilization adaptively. The random approach distributes connections without following a specific pattern, providing insight into the impact of randomness on scalability. Meanwhile, the Round Robin approach distributes connections evenly and cyclically among the servers, ensuring that all receive a similar number of connections over time.

As illustrated in Figure 3, the loading time of the dynamic approach proved to be considerably faster compared to the other two approaches. This result highlights the effectiveness of selecting the least occupied server, which is the main characteristic of the dynamic approach as proposed by this load balancing solution.

The ability to dynamically adjust the distribution of connections based on the current state of the servers allowed for more efficient use of available resources, resulting in faster response times and better overall system performance.

Figure 4 presents the results of bitrate variation during video playback in this scenario. Once again, the dynamic approach proved to be more consistent, providing a higher quality of experience more quickly compared to the other approaches. In the dynamic approach, selecting the least occupied server allowed for more efficient data delivery, resulting in a more stable and high-quality video experience.

On the other hand, the random approach showed points of quality loss, highlighting its lack of efficiency in load

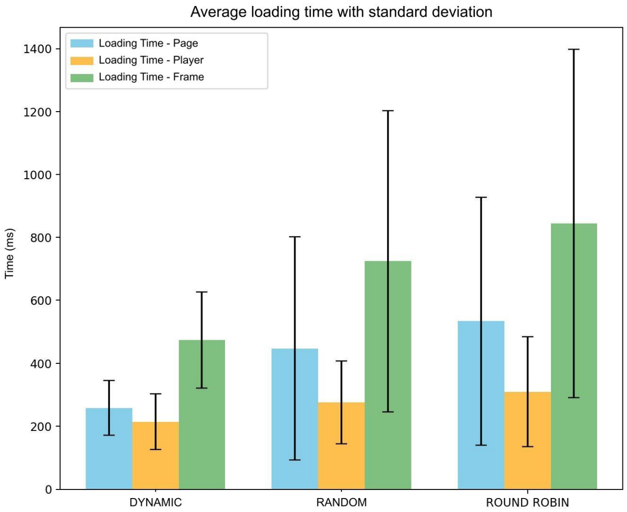

Fig. 3. Response time - scalability test 100 10

distribution. The random distribution of connections led to uneven server utilization, resulting in variations in bitrate and, consequently, a less satisfying video playback experience. None of the evaluated approaches presented stalls.

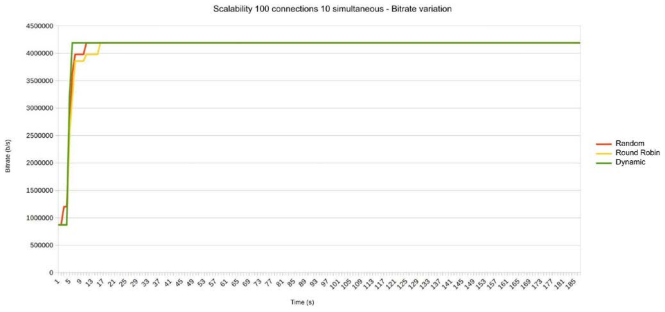

Fig. 4. Bitrate variation - scalability test 100 10

In the second scenario, every 5 seconds, a random server in the network receives 1000 connections, with a maximum of 100 connections being simultaneous. This substantial increase in load aims to evaluate how each load balancing approach handles high-demand conditions, testing the limits of the infrastructure.

Figure 5 illustrates the loading times for all approaches tested in the 1000-connection scenario. The dynamic approach again proved to be the most consistent and fastest compared to the other two approaches.

The faster loading times of the dynamic approach highlight its effectiveness in adjusting connection allocation based on the current state of the servers, allowing for better load distribution and more efficient resource utilization. In contrast, the random and Round Robin approaches showed greater variations in loading times, indicating less efficiency in managing the load under high-demand conditions.

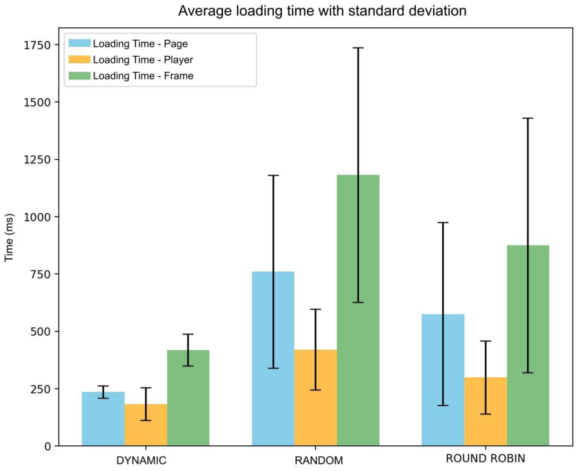

Fig. 5. Response time - scalability test 1000 100

Figure 6 shows that there was no significant variation in bitrate throughout the video playback for the different approaches tested. However, a slight advantage of the dynamic approach is noticeable, as it achieves the best quality more quickly compared to the other approaches. None of the evaluated approaches experienced stalls.

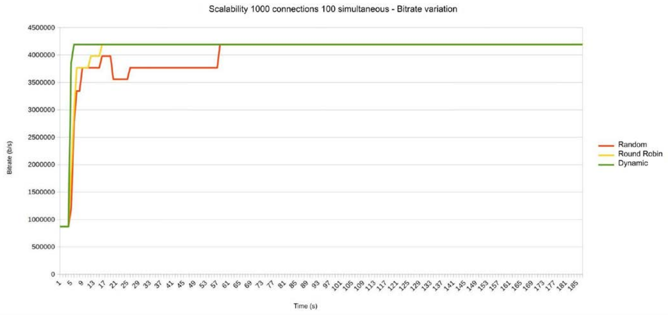

Fig. 6. Bitrate variation - scalability test 1000 100

These results reinforce the superiority of the dynamic approach in high-performance environments, where speed and consistency in response time are crucial for maintaining service quality and user experience.

The third scalability test aims to saturate the network to a degree that potentially impacts user experience quality. In this scenario, every 5 seconds, a random server receives up to 5000 connections, with a maximum of 500 connections being simultaneous. This test was designed to examine how each approach handles extreme high-demand conditions, pushing the infrastructure beyond to its limits.

Figure 7 illustrates the loading times, again showing that the dynamic approach remains more efficient than the other approaches. In extreme high-demand scenarios, the dynamic approach's ability to adjust connection distribution in real time results in faster and more consistent response times, highlighting its superiority under extreme load conditions.

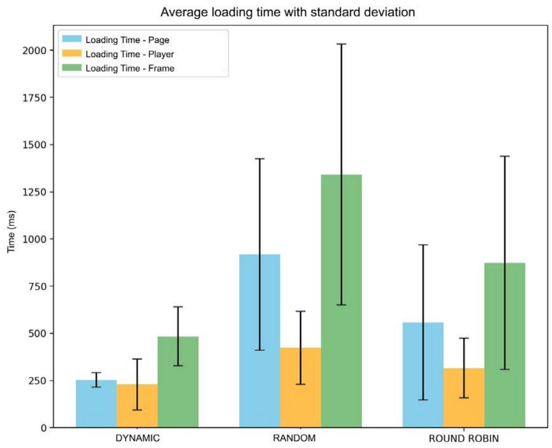

Fig. 7. Response time - scalability test 5000 500

As shown in Figure 8, the bitrate variation was more inconsistent in this test scenario, reflecting the impact of high network saturation. However, the dynamic approach managed to maintain maximum quality throughout the test period. This consistency in bitrate quality highlights the dynamic approach's efficiency in managing load and optimizing data delivery, ensuring a superior user experience even under extreme stress conditions. None of the evaluated approaches experienced stalls.

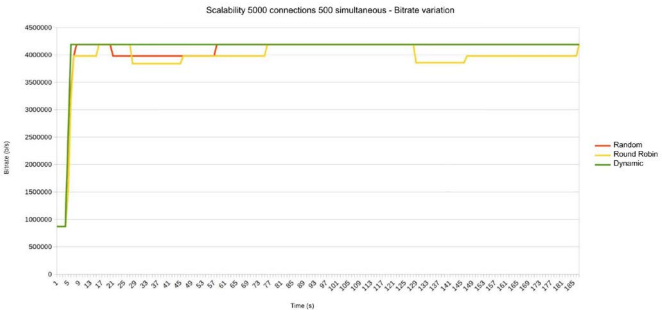

Fig. 8. Bitrate variation - scalability test 5000 500

These results confirm that, during extreme peak conditions, the dynamic approach is not only more efficient in loading times but also maintains service quality at high levels, solidifying its effectiveness in high-demand environments.

V. Conclusion

This study aims to provide a viable solution, or at least a significant mitigation, to the challenge of optimizing video streaming in content delivery networks. In this context, it proposes the implementation of a video-streaming server's load-balancing system based on Software-Defined Networking (SDN) to intercept MPEG-DASH video packets during video playback. The system analyzes the performance metrics of available content servers and, based on these analyses, carefully selects the most appropriate server to handle client content requests.

The results obtained with this approach are promising. In all situations where time was the evaluated metric, the approach proved to be more agile and consistent, as evidenced by the low standard deviation of the results. Additionally, in tests focused on observing bitrate fluctuations, there was notable consistency in content playback. Once maximum quality was achieved, it remained stable until the end of the video playback. Furthermore, even with the dynamic redirection of connections during streaming, no perceptible changes in video quality were observed. This stability is crucial for ensuring a satisfactory user experience, and use in real world scenarios.

In summary, applying SDN for video streaming optimization proves to be an effective strategy. Dynamic and intelligent server selection based on performance metrics can significantly enhance user experience and network efficiency, making it a promising alternative for the challenges faced in multimedia content distribution.

Acknowledgments

This work received financial support from the Coordination for the Improvement of Higher Education Personnel - CAPES - Brazil (PROAP/AUXPE).

References

[1] I. Global Industry Analysts, "Cloud computing services - global market trajectory & analytics," 2021.

[2] J. Stoll, "Ott video revenue worldwide from 2010 to 2026," 2021.

[3] Sandvine, "Global internet phenomena," 2023. [Online]. Available: https://www.sandvine.com/phenomena

[4] M. Haris and R. Z. Khan, "A systematic review on load balancing tools and techniques in cloud computing," in *Inventive Systems and Control*, V. Suma, Z. Baig, S. Kolandapalayam Shanmugam, and P. Lorenz, Eds. Singapore: Springer Nature Singapore, 2022, pp. 503–521.

[5] D. Liu, Z. Wang, and J. Zhang, "Video stream distribution scheme based on edge computing network and user interest content model," *IEEE Access*, vol. 8, pp. 30 734–30 744, 2020.

[6] R. H. Majdabadi, M. Wang, and L. Rakai, "Soda-stream: Sdn optimization for enhancing qoe in dash streaming," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, 2022, pp. 1–5.

[7] D.-I. I. W. Group, "Content steering for dash," 2022, disponível em https://dashif.org/docs/DASH-IF-CTS-00XX-Content-Steering-Community-Review.pdf. Acesso 8 Ago. de 2024.

[8] T. S. Andjamba and G.-A. L. Zodi, "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)*, 2023, pp. 1–6.

[9] M. Taha, "An efficient software defined network controller based routing adaptation for enhancing qoe of multimedia streaming service," *Multimedia Tools and Applications*, vol. 82, pp. 1–24, 03 2023.

[10] A. M. Bamhdi, "Cdca: Transparent cache architecture to improve content delivery by internet service providers," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, 2023. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2023.0141090

[11] S. M. A. H. Bukhari, M. Afaq, and W.-C. Song, "Streaming via sdn: Resource forecasting for video streaming in a software-defined network," in *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2023, pp. 596–601.

[12] W.-K. Chiang and T.-Y. Li, "An extended sdn architecture for video-on-demand caching," *Mobile Networks and Applications*, pp. 1–18, 04 2024.

[13] CloudLab, "The cloudlab manual," 2023, acesso em: 10 out. 2023. [Online]. Available: https://docs.cloudlab.us/