Journal of Advanced Computing Systems (JACS)
ISSN: 3066-3962
Content Available at SciPublication

Reinforcement Learning for Efficient and Low-Latency Video Content Delivery: Bridging Edge Computing and Adaptive Optimization

Zhuolin Ji [1], Chenyu Hu[1.2], Gengrui Wei[2]*

[1] *Master of Computer Vision & Control, Illinois institute of technology,IL,USA*
[1.2] *Digital Social Media, University of Southern California, CA, USA*
[2] *Computational Science and Engineering, Virginia Tech, VA, USA*
**Corresponding author E-mail: eva499175@gmail.com*

DOI: 10.69987/JACS.2024.41205

Keywords

Video Transmission;
Reinforcement Learning;
EC (edge computing);
Adaptive Optimization

Abstract

In 2019, video transmission traffic made up 60.6% of overall Internet downlink traffic. In the future, with the rapid development of 4K/8K, AR/VR, holographic communication, smart city, intelligent transportation, and other technologies, network video transmission demand and traffic will be further inspired. In addition, the number of video users on the Internet has maintained a rapid growth tendency, not only due to the rapid improvement of traditional network bandwidth but also because the quick expansion of mobile Internet has further stimulated the potential of the video transmission market. This paper designs a video transmission optimization strategy that takes reinforcement learning and edge computing (TORE) to improve the video transmission efficiency and quality of experience. Specifically, first, we design the popularity prediction model for video requests based on the RL (reinforcement learning) and introduce the adaptive video encoding method for optimizing the efficiency of computing resource distribution. Second, we design a video caching strategy, which adopts EC (edge computing) to reduce the redundant video transmission. Last, simulations are conducted, and the experimental results fully demonstrate the improvement of video quality and response time.

1. Introduction

With the rapid development of Internet technology, the global demand for video content has surged, driven by applications such as real-time streaming, live broadcasting, and online gaming. These applications require low latency and high bandwidth, posing significant challenges to traditional network architectures. The best-effort packet forwarding model of the Internet struggles to meet these demands due to issues like transmission redundancy, inefficient resource utilization, and lack of coordination between end-users and network components. [1]To address these challenges, researchers have explored various optimization approaches, including Information-Centric Networking (ICN) and Content Delivery Networks (CDNs). However, these solutions face limitations such as high deployment costs, limited coverage, and an inability to adapt dynamically to network conditions.

Recent advancements in adaptive bitrate (ABR) algorithms and edge computing have shown promise in improving video transmission efficiency and user Quality of Experience (QoE). ABR algorithms dynamically adjust video bitrates based on available bandwidth, reducing buffering and improving user satisfaction. However, traditional ABR strategies often focus on local optimization, which may not guarantee global network resource utilization. Meanwhile, edge computing platforms, located closer to end-users, enable global optimization of video delivery by leveraging enhanced sensing, storage, and computational capabilities. These platforms can address the limitations of traditional CDNs and ABR algorithms, providing real-time adaptation to network dynamics and improving overall QoE.

In this context, this paper proposes a novel video content distribution optimization framework based on **reinforcement learning (RL)[2]**. Unlike traditional static optimization methods, RL enables continuous learning and adaptation, allowing for real-time

adjustments to content distribution strategies. The primary objective of this research is to design an intelligent CDN optimization framework that improves video transmission efficiency, reduces latency, and enhances user QoE, particularly under high-load and complex network conditions. By integrating reinforcement learning with edge computing, this work bridges the gap between adaptive optimization and global resource utilization, offering a scalable and dynamic solution for modern video delivery challenges.

Related Work

2.1. Optimization of Transmission Architecture

To enhance network transmission efficiency, researchers have developed various optimization algorithms targeting the limitations of the traditional best-effort packet forwarding mode. These algorithms aim to address key issues such as transmission redundancy and the lack of coordination between end-side and network-side operations. The ultimate goal is to create a comprehensive video transmission optimization scheme that leverages advancements in network architecture and transmission protocols.

One significant development in this area is the Content-Centric Network (CCN), proposed by Lv in 2009 as a novel Internet architecture that shifts the focus from host-centric to content-centric communication. CCN is a prominent example of Information-Centric Networking (ICN), which redefines network communication by prioritizing information over traditional IP-based packet switching. ICN introduces mechanisms like content labeling and network caching to minimize redundancy and enhance the efficiency of video transmission. Among the various ICN implementations, CCN stands out due to its robust naming, routing, distribution, and caching mechanisms [3-4].

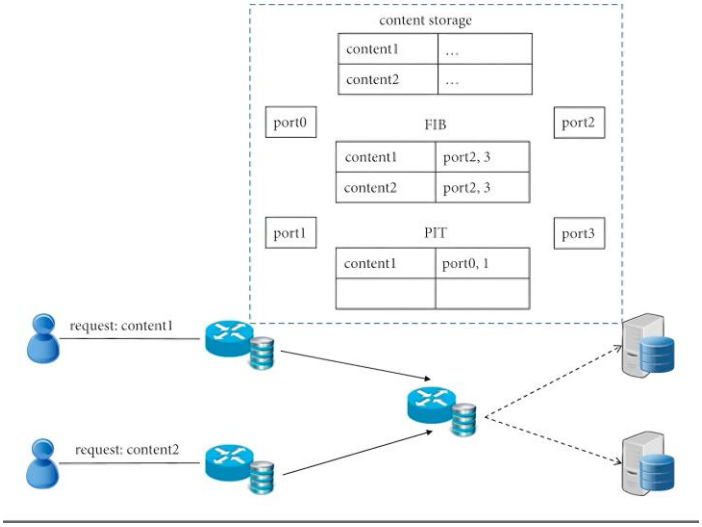

Figure 1. illustration of CCN routing mechanism.

Key technologies in CCN include name-based content routing, facilitated by the Forwarding Information Base (FIB) and Pending Interest Table (PIT), and network caching, which allows routers to store content and employ diverse replacement strategies. Recent advancements in CCN and ICN have further optimized these mechanisms, integrating machine learning for intelligent caching and adaptive routing, thereby pushing the boundaries of network efficiency and scalability. One of the key technologies of CCN is name-based content routing[5]. The implementation of name-based content routing includes two main modules: forwarding information base (FIB) and pending interest table (PIT), which are shown in Figure 1. FIB is able to forward interest messages to nodes that may cache corresponding data. Another key technology of CCN is network caching. The router in CCN can achieve a content storage module (as shown in Figure 1), which is similar to the buffer space in the IP network, but it can have different content replacement strategies.

2.1.1. Content Distribution Network

Related schemes of Information-Centric Networking (ICN) have shown potential in improving the efficiency of network video transmission. However, their implementation in the current Internet infrastructure

remains challenging in the short term. As a result, application-layer solutions like Content Distribution Networks (CDNs) have become the dominant approach for optimizing video transmission. CDNs are widely adopted by content service providers globally, including Akamai in the United States, as well as domestic providers like Netresidence Technology and Blue Flood in China. Additionally, major cloud service providers such as Alibaba Cloud and Tencent offer robust CDN-based content acceleration services.

CDNs function as platforms for accelerated content distribution, enabling content providers (CPs) to deliver video and other media efficiently. CDN service providers deploy multiple cloud platforms and Points of Presence (PoPs) across regions or even globally. By pre-positioning content (e.g., high-volume video files) at these PoPs, CDNs ensure that user requests are redirected to the nearest PoP through DNS-based canonical name resolution[6]. This approach significantly reduces the bandwidth demands on the backbone network and minimizes redundant data transmission between CDN cloud platforms and PoPs, thereby enhancing overall network transmission efficiency.

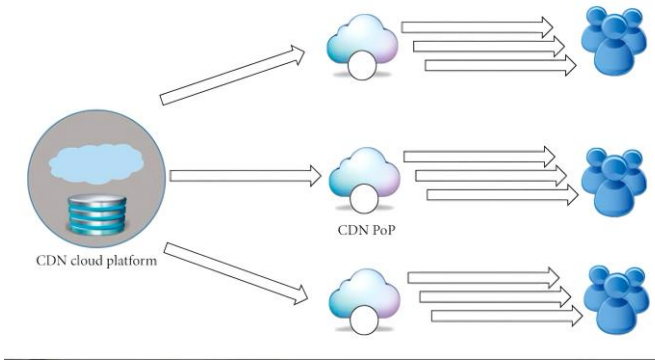

Figure 2. Illustration of CDN architecture

For instance, platforms like YouTube, which handle massive video traffic, rely heavily on CDNs to deliver content seamlessly to billions of users worldwide. YouTube's video transmission strategy leverages CDN infrastructure to pre-cache popular videos at edge servers, ensuring low latency and high-quality streaming. This not only reduces the load on YouTube's central servers but also optimizes bandwidth usage across the network. As video consumption continues to grow, CDNs play a critical role in meeting the increasing demand for efficient and scalable video transmission.As shown in Figure 2, from the perspective of content transmission, CDN can greatly decrease the bandwidth requirements of the backbone network and eliminate huge redundant transmission between CDN cloud platforms and CDN PoPs, so as to improve the network transmission efficiency[7]. In addition, CDN is closed and independent based on the application layer, where content service providers and network service providers cannot participate in the optimization of content distribution, so the available communication and joint optimization mode between the network side and the end side are impossible to form.

2.2 Video Transmission Optimization Algorithm

The rapid growth of video traffic, driven by platforms like YouTube, Netflix, and TikTok, has made optimizing video transmission essential for maintaining high-quality user experiences. Video traffic now dominates global internet usage, accounting for over 60% of total bandwidth, with trends pointing toward further increases due to the adoption of higher-resolution formats like 4K and 8K. To address the challenges of dynamic network conditions and diverse user demands, researchers have developed advanced optimization algorithms, including adaptive bitrate adjustment and edge computing-based solutions.

A key approach to improving video transmission is the Adaptive Bitrate (ABR) algorithm, which dynamically adjusts video quality based on real-time network conditions to minimize buffering and enhance user Quality of Experience (QoE). ABR algorithms aim to balance video quality and playback smoothness by adapting to available bandwidth. However, configuring ABR parameters to account for varying network states and user system characteristics remains a significant challenge. To address this, MIT's research team introduced Pensieve, a reinforcement learning-based ABR scheme[8]. As shown in Figure 2, Pensieve leverages machine learning to intelligently adjust video bitrates, eliminating the need for manual parameter tuning and demonstrating superior performance in real-

world applications. Despite its advancements, end-based ABR strategies like Pensieve are limited by their local optimization focus, as each client independently adjusts its request policy without considering the global utilization of network bandwidth resources.

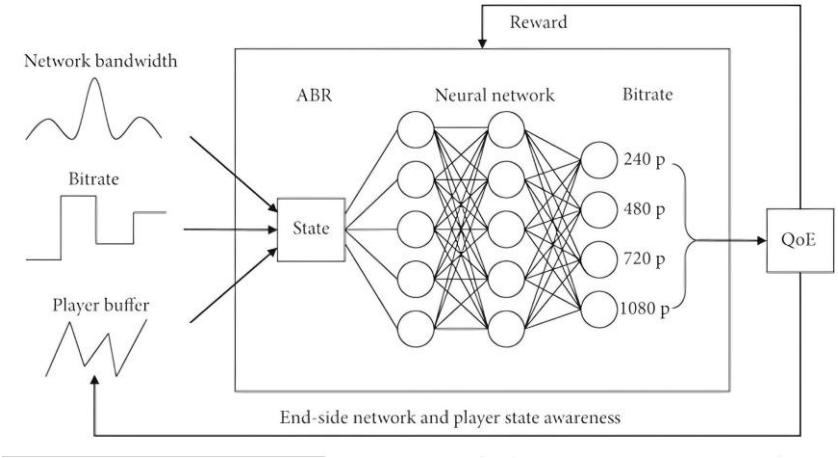

Figure 3. Reinforcement learning-based intelligent pensive-ABR mechanism.

To overcome these limitations, edge computing has emerged as a transformative solution for intelligent video transmission. Edge computing platforms, positioned close to end-users, enable global optimization of video transmission for multiple users sharing bottleneck bandwidth. These platforms leverage their sensing, storage, and computational capabilities to reduce transmission redundancy, improve bandwidth utilization, and mitigate dynamic network jitter. Recent studies have proposed joint bitrate optimization mechanisms based on edge computing, utilizing deep learning to make intelligent decisions.

These schemes outperform traditional end-based QoE optimization methods by achieving higher overall QoE and more efficient resource allocation. For example, edge computing can pre-process and cache popular video content at edge servers, as illustrated in Figure 2, reducing latency and bandwidth demands on the core network. This is particularly beneficial for live streaming and real-time video applications, where low latency and high reliability are critical. By combining local adaptability with global resource management, edge computing and ABR algorithms represent a promising direction for optimizing video transmission in the face of ever-growing traffic demands.In literature [9-10], the authors proposed joint bitrate optimization mechanisms based on edge computing. These schemes make intelligent joint bitrate decisions through deep learning. Compared with the traditional end-based QoE optimization mechanism, the optimization scheme based on edge computing has prominent advantages in terms of total QoE.

Video Transmission Optimization Based on RL and EC

3.1. Cloud-Based Intelligent Video Coding Mechanism

The video is increasingly popular as a core experience of people's online activities. Only on Facebook, more than 8 billion videos are viewed every day. The client downloads videos from the cloud server of the video provider by ABR to watch videos. The ABR algorithm can dynamically select the highest bitrate that the network bandwidth can support and avoid the jam phenomenon during watching. Higher bitrate can provide higher video quality, but it also results in more video transmissions, so the end-to-end connection with the higher bandwidth is required for clients.

When the original videos are uploaded, different basic bitrate versions of the videos are generated [11], which consumes huge computing resources. In the network video transmission, there are more than 100 video resolutions, and the same resolution also contains multiple different video bitrates, so the number of potential output types of video bitrates is large. By default, FFmpeg is used to encode the video uploaded to the server into a small number of standard versions. More computation can improve the user's video viewing experience by improving the coding performance (decreasing the amount of transmitted data for the same video quality) or increasing the coding selection (providing more fine-grained bitrate selection to adapt

to the dynamic network bandwidth). However, the computing power of video coding in the cloud is limited, and it is impossible to generate enough coding versions for all videos. Therefore, dynamically allocating appropriate coding power to the cloud among different videos to achieve the optimal global user experience is one of the problems to be solved in the network video transmission.

In this paper, we propose a cloud-based intelligent video coding mechanism with popularity consideration, assigning computing power and encoding bitrate versions of videos according to the popularity. However, the popularity of videos in the real situation is extremely imbalanced, where less than 1% of the videos contribute more than 80% of the time spent in viewing, so the imbalance is very obvious. This feature is of great value to computing power allocation for cloud dynamic coding. In the cloud, the highest quality coding or more customized bitrate versions are produced on demand for a small number of the most popular videos, so that the overall video viewing quality can be significantly improved with only a small amount of computing power.

3.1.1. Prediction of Video Popularity Based on Reinforcement Learning

Analysis and prediction of video popularity are required for targeting cloud coding based on the feature of high concentration of video watching. In our scheme, the request processing logging mode is in charge of logging the sequence of video user requests, including video ID, request bitrate, request time, terminal parameters (such as resolution), etc. The popularity prediction should have following characteristics: first, the prediction should be quick, so that it can decrease the number of missing video requests; second, the prediction should be accurate, which can ensure that the computing is consumed on the most valuable videos; and third, the prediction should be scalable to analyze and predict massive request records.

The popularity prediction methods proposed in papers [12-14] mainly aimed at the analysis and prediction of popularity at the day level. These methods need great prediction delay, and the goal of this paper is to quickly predict the popularity at the minute level, so it is very important to design a fast-incremental popularity prediction algorithm. To be able to further maintain stability and adaptability to network dynamics, we use reinforcement learning to predict the popularity of videos.

Video requests that occurred in the past time t will have an impact on the popularity of future moment T, which is represented by f (T-t). f is a function of probability distribution defined on the space [0, +∞], which is generally monotonically decreasing. Therefore, in principle, the more recent the visit, the greater the effect on the popularity, and the effect of a particular visit on the popularity gradually converges to zero over time. For a video, tirepresents the time of the visit i; and the total number of times to watch the video in the future time T can be calculated by the following formula:

$$F(T) = \sum_{t_i \le T} \int_{t}^{+\infty} f(t - t_i)\, dt. \quad (1)$$

The key problem is to set the core probability density function f to make incremental update possible, so as to accelerate the process of video popularity prediction. Previous works [15] used power law distribution as the probability density function to predict the popularity. However, a complete calculation is required to solve the popularity every time in this method, which greatly decreases the prediction speed and affects the timeliness of the popularity feedback. In this paper, we use exponential distribution as the probability density function, which can largely reduce the computations needed for the popularity prediction, and is expressed as follows:

$$f(t) = \left(\frac{1}{w}\right) e^{-(t/w)}, \quad (2)$$

where w indicates the range of the time window for future impact, and it mainly serves to remove visits made long ago, which have minimal effect on the accuracy of popularity prediction and can be ignored. For a video, we suppose T2 is the present request time of the video to trigger the present popularity upgrade, and T1 is the last request time of the request.

Experimental Simulation and Result Analysis

4.1. Setups

To prove the effectiveness and efficiency of the mechanism, simulations are conducted based on four parts: video data, mechanism settings, video requests, and comparison simulations.

4.1.1. Video Data

We use 1000 videos for simulations, and 25 new videos will incrementally be uploaded every 1 s during the experiments. For each video, 10 blocks are contained, and the playing time of each block is 2 s.

4.1.2. Mechanism Settings

(1) Computing power model setting: CPU computing power is set to 400 cores. With a single-core CPU computing power, the video encoding time for each bitrate is uniformly set to 5 s, which means that the cloud computing power could handle 80 video encoding missions at a second.

(2) Regular transcoding power distribution setting: The encoding range of the video is considered {180p, 240p, 360p, 480p, 540p, 720p, 960p, 1080p}. All of these, except for the regular coding, are used as potential on-demand custom coding requirements, triggered by the popularity of user requests. By default, the original videos are encoded as 360p and 720p bitrates. Since the beginning of the experiment, the regular encoding of all new videos is required, and this approach allocates 1/2 of the computing power to regular encoding.

(3) Edge computing platform: The cache capacity is configured depending on the storage capacity of 400 videos with the bitrate of 180p. If 360p is targeted, the platform is able to cache 200 videos, and so on. The transmission latency between the edge and the user is 5 ms, and the transmission latency from the cloud to the edge platform is 200 ms.

(4) Network bandwidth setting: Assuming that no bandwidth bottleneck exists between the edge and the user, and the downlink traffic between the cloud video platform and the edge platform can transmit 400 video blocks (each video has 10 video blocks) with the bitrate of 480p per second. For 960p, only 200 video blocks can be completed per second, and so on.

4.1.3. Video Requests

Video requests distribution setting: The user chooses a video according to the Zipf (parameter 1.07) probability distribution to request and randomly chooses a bitrate from {180p, 240p, 360p, 480p, 540p, 720p, 960p, 1080p}.

4.1.4. Comparison Simulations

Comparison simulations are conducted among our mechanism, joint coding-transmission optimization (TOSO) [29], and joint rate control and buffer management (JRCBM) [16], under different numbers of requests, and the results are analyzed in terms of video relative quality and video lag degree.

4.1.5. Data Table for Simulation Parameters

To provide a clear overview of the simulation setup, Table 1 summarizes the key parameters used in the experiments.

Parameter	Value
Number of Videos	1000 (with 25 new videos uploaded every 1 second)
Blocks per Video	10 blocks (each block has a playing time of 2 seconds)
CPU Computing Power	400 cores (5 seconds per bitrate encoding, 80 missions per second)
Encoding Bitrates	{180p, 240p, 360p, 480p, 540p, 720p, 960p, 1080p}
Default Encoding Bitrates	360p and 720p
Edge Cache Capacity	400 videos at 180p, 200 videos at 360p, etc.
Transmission Latency	Edge to User: 5 ms, Cloud to Edge: 200 ms
Network Bandwidth	400 video blocks (480p) or 200 video blocks (960p) per second
Video Request Distribution	Zipf distribution (parameter 1.07) with random bitrate selection

Table 1. *Summary of Simulation Parameters.*

4.2. Results Analysis

4.2.1. Video Relative Quality

The comparison simulations on video relative quality under different numbers of requests are shown in Figure 4.

Our algorithm is always the best under different numbers of requests because, when the basic bitrates do not

match the user request, the coding task can be customized to ensure the relative quality of the video.

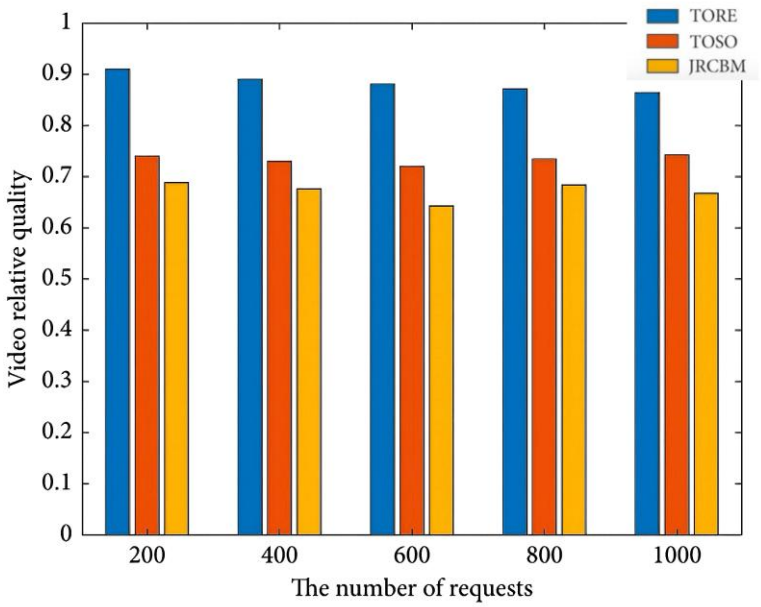

Figure 4. Comparison on video lag degree under different numbers of requests.

4.2.3. Video Response Time

As can be seen from Figure 8, the proposed TORE has a good performance in response time. The intelligent

caching method is implemented according to the regional popularity characteristics in the EC platform, which is

combined with video forwarding to minimize the network transmission redundancy and maximize the video

transmission efficiency. The proposed scheme is of significant value for optimizing the video response time,

which can improve the network transmission efficiency and user QoE[17].

Comparison simulations on the video lag degree under different request numbers are shown in Figure 5, and the

proposed TORE is always the best under different numbers of requests. We can explain the advantages of the

proposed approach in two aspects. On one hand, the EC-based intelligent caching strategy adaptively allocates

arithmetic power and tasks to edge-side nodes, which will decrease the transmission latency of the requests. On

the other hand, the popularity-based edge intelligent caching reduces the redundant transmission of the network.

As a result, the path will not be jammed to ensure the stability of the huge network video transmission.

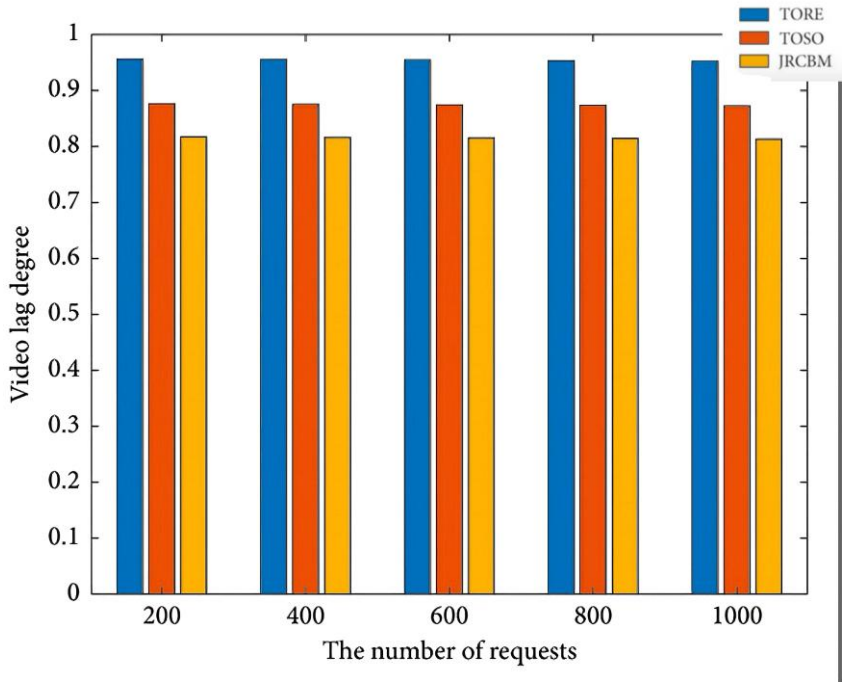

Figure 5. Comparison on video lag degree under different numbers of requests.

4.2.3. Video Response Time

As can be seen from Figure 6, the proposed TORE has a good performance in response time. The intelligent caching method is implemented according to the regional popularity characteristics in the EC platform, which is combined with video forwarding to minimize the network transmission redundancy and maximize the video transmission efficiency. The proposed scheme is of significant value for optimizing the video response time, which can improve the network transmission efficiency and user QoE.

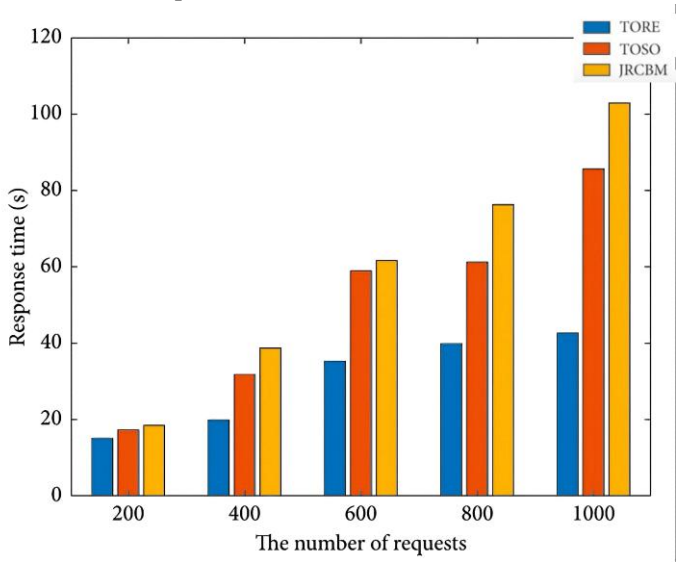

Figure 6. Comparison on video response time under different numbers of requests.

4.3. Discussion

The experimental results demonstrate the effectiveness of the proposed mechanism in optimizing video relative quality, reducing video lag degree, and improving video response time. As shown in Figure 6, the proposed TORE mechanism consistently outperforms TOSO and JRCBM in terms of response time[18-19]. This is primarily due to the intelligent caching strategy implemented in the edge computing (EC) platform, which leverages regional popularity characteristics to minimize network transmission redundancy and maximize video transmission efficiency. By adaptively allocating computing resources and tasks to edge-side nodes, the proposed approach significantly reduces transmission latency and ensures stable video delivery even under high request loads.

Furthermore, the popularity-based edge caching strategy plays a crucial role in reducing redundant network transmissions, which not only alleviates network congestion but also enhances the overall user Quality of Experience (QoE). The results highlight the importance of integrating intelligent caching and adaptive resource allocation in modern video delivery systems, especially in scenarios with dynamic user demands and varying network conditions. Future work could explore the scalability of the proposed mechanism in larger networks and its applicability to real-time video streaming applications with stricter latency requirements.

1. CONCLUSION

In this paper, we propose a dynamic computing power allocation mechanism based on intelligent popularity prediction for video user distribution. The proposed mechanism effectively balances conventional encoding demand and dynamic on-demand customized encoding requests, enabling the efficient and adaptive allocation of limited cloud computing resources. By integrating reinforcement learning and edge computing, our method optimizes the distribution of computing power among servers, ultimately reducing response latency and enhancing the Quality of Experience (QoE) for users[20]. Through experimental validation, we demonstrate that our approach improves the efficiency of video transmission while minimizing network latency, making it a promising solution for modern video content delivery systems.

A key contribution of our proposed optimization mechanism lies in its ability to enhance video quality and response time, ensuring a seamless viewing experience for users. By leveraging intelligent video popularity prediction, we achieve adaptive caching, thereby further reducing network congestion and computational overhead. However, while our research focuses on optimizing computing resource allocation and network efficiency, it does not explicitly analyze the impact of video compression and decoding on transmission performance. Given that these aspects also play a crucial role in determining the overall quality and efficiency of video delivery, future work could explore their integration within our proposed framework.

Looking ahead, further optimizations could be introduced to refine video content delivery. One potential avenue involves implementing adaptive bitrate encoding techniques to adjust video quality based on user engagement levels. For instance, videos deemed less relevant to users could be encoded at lower bitrates, effectively reducing redundant traffic without compromising the overall user experience. By incorporating such strategies, future research could extend the benefits of our proposed model, achieving even greater efficiency in video streaming while maintaining high levels of responsiveness and quality.

2. Acknowledgment

The authors would like to express their sincere gratitude to all colleagues and collaborators who contributed valuable insights and discussions throughout the development of this research[13]. Special thanks are extended to the editorial team and reviewers of the Academic Journal of Science and Technology*for their constructive feedback, which has significantly improved the quality of this work. This research was supported in part by [relevant funding agency, if applicable], and we appreciate their generous support in advancing deep learning-based image segmentation studies.

We extend our heartfelt appreciation to our research team and academic peers for their invaluable contributions and constructive discussions that helped refine this study[14]. Additionally, we thank the organizers of the 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE) for providing a platform to present and exchange ideas on optimizing data flow in machine learning models. Finally, we acknowledge any institutional or funding support that facilitated this research, enabling us to explore the integration of AutoML in data pipeline training.

REFERENCES

[1]. Raman, Adithya, Bekir Turkkan, and Tevfik Kosar. "LL-GABR: Energy Efficient Live Video Streaming Using Reinforcement Learning." arXiv preprint arXiv:2402.09392 (2024).

[2]. Spielberg, S. P. K., R. B. Gopaluni, and P. D. Loewen. "Deep reinforcement learning approaches for process control." 2017 6th international symposium on advanced control of industrial processes (AdCONIP). IEEE, 2017.

[3]. Elharrouss, Omar, et al. "Backbones-review: Feature extractor networks for deep learning and deep reinforcement learning approaches in computer vision." Computer Science Review53 (2024): 100645.

[4]. Ansari, Yasmeen, et al. "A deep reinforcement learning-based decision support system for automated stock market trading." IEEE Access 10 (2022): 127469-127501.

[5]. Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simulta- neous localization and mapping problem. Aaai/iaai, 593598, 2002.

[6]. Wang, P., Varvello, M., Ni, C., Yu, R., & Kuzmanovic, A. (2021, May). Web-lego: trading content strictness for faster webpages. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications (pp. 1-10). IEEE.

[7]. Ni, C., Zhang, C., Lu, W., Wang, H., & Wu, J. (2024). Enabling Intelligent Decision Making and Optimization in Enterprises through Data Pipelines.

[8]. Zhang, C., Lu, W., Ni, C., Wang, H., & Wu, J. (2024, June). Enhanced user interaction in operating systems through machine learning language models. In International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024) (Vol. 13180, pp. 1623-1630). SPIE.

[9]. Wang, H., Wu, J., Zhang, C., Lu, W., & Ni, C. (2024). Intelligent security detection and defense in operating systems based on deep learning. International Journal of Computer Science and Information Technology, 2(1), 359-367.

[10]. Lu, W., Ni, C., Wang, H., Wu, J., & Zhang, C. (2024). Machine learning-based automatic fault diagnosis method for operating systems.

[11]. Zhang, C., Lu, W., Wu, J., Ni, C., & Wang, H. (2024). SegNet network architecture for deep learning image segmentation and its integrated applications and prospects. Academic Journal of Science and Technology, 9(2), 224-229.

[12]. Wu, J., Wang, H., Ni, C., Zhang, C., & Lu, W. (2024, March). Data Pipeline Training: Integrating AutoML to Optimize the Data Flow of Machine Learning Models. In 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 730-734). IEEE.

[13]. Wu, J., Wang, H., Ni, C., Zhang, C., & Lu, W. (2024). Case Study of Next-Generation Artificial Intelligence in Medical Image Diagnosis Based on Cloud Computing. Journal of Theory and Practice of Engineering Science, 4(02), 66-73.

[14]. Ni, C., Wu, J., Wang, H., Lu, W., & Zhang, C. (2024, June). Enhancing cloud-based large language model processing with elasticsearch and transformer models. In International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024) (Vol. 13180, pp. 1648-1654). SPIE.

[15]. Huang, D., Yang, M., & Zheng, W. (2024). Using Deep Reinforcement Learning for Optimizing Process Parameters in CHO Cell Cultures for Monoclonal Antibody Production. Artificial Intelligence and Machine Learning Review, 5(3), 12-27.

[16]. Bi, W., Trinh, T. K., & Fan, S. (2024). Machine Learning-Based Pattern Recognition for Anti-Money Laundering in Banking Systems. Journal of Advanced Computing Systems, 4(11), 30-41.

[17]. Ma, X., & Fan, S. (2024). Research on Cross-national Customer Churn Prediction Model for Biopharmaceutical Products Based on LSTM-Attention Mechanism. Academia Nexus Journal, 3(3).