This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

Multipath Smart Preloading Algorithms in Short Video Peer-to-Peer CDN Transmission Architecture

Dehui Wei[1], Jiao Zhang [1 2 *], Haozhe Li[1], Zhichen Xue[3], Yajie Peng[3], and Rui Han[3]

[1]State Key Laboratory of Networking and Switching Technology, BUPT, China. Email: jiaozhang@bupt.edu.cn.
[2]Purple Mountain Laboratories, Nanjing, China
[3]ByteDance Ltd. Beijing

***Abstract*—The rapid growth of short video users has brought high traffic costs to content providers. Saving Content Distribution Network (CDN) costs while maintaining users' playback Quality of Experience (QoE) is a significant problem for major short video platforms like Douyin (the Chinese counterpart of TikTok). Two commonly utilized cost-saving methods: PCDN (peer-to-peer CDN, fetching data from multiple low-cost but low-bandwidth edge devices with a multipath transmission protocol) and preloading control (minimizing unnecessary future playing data by downloading videos in segmented data ranges), each may hurt users' experience respectively. Worse still, both methods combined have a synergistic negative effect over QoE. In this paper, we focus on experiences, algorithms, and prospects to solve this cost-QoE dilemma. We first introduce Douyin's current PCDN multipath architecture and then review learning-based preloading techniques. Second, based on Reinforcement Learning (RL), we propose a Multipath-aware Smart Preloading algorithm, which consists of three schemes: one to decide the best size of the next range of preloaded data, another to design a water level valve algorithm that prioritizes preloading between currently playing video's unfinished data and the next video's beginning data, and the last one to determine the bitrate level of the next video. Douyin's anonymous user feedback shows our Smart Preloading algorithm reduces traffic waste by ∼26% while ensuring QoE. Third, we analyze and outlook the future of video systems, including trends in PCDN and other open issues.**

***Index Terms*—Preloading, Reinforcement Learning, PCDN, Multipath transmission, Short video**

I. Introduction

As a novel form of content interaction, short videos have garnered a substantial user base since their inception [1]. For instance, Douyin has over 1 billion daily active users and an annual traffic cost exceeding 6 billion RMB. To ensure smooth playback for users, content providers typically distribute video data through CDN [2]. The high traffic costs have necessitated the transformation of the video transmission architecture to PCDN, a low-cost and high-quality content distribution network service built by the massive fragmented idle resources of the edge network.

Preloading control is another major cost-saving method. Preloading is necessary for a better user experience in short video playback, which usually involves in-video preloading and cross-video preloading. In-video preloading downloads future data of the current playing video for users to manually drag the progress bar to a farther position, and cross-video preloading enables advanced downloads of recommended videos while playing the current video, facilitating faster and smoother transitions between videos. In Douyin's early experience, the entire video was preloaded to ensure optimal user experience. However, this approach resulted in wasted traffic if users slid away or exited from watching the current video. Downloading a small piece of data behind the user's viewing position instead of the entire video can significantly reduce costs but may compromise QoE. Each preloading segment of a video is called a range in this paper. Preloading control is a mechanism for managing the balance between preloaded data waste and user experience, which requires careful management of range size in both in-video and cross-video scenarios. However, optimizing preloading control can be challenging [3] due to the elusive nature of user behavior and the dynamic network conditions that make it difficult to determine appropriate preload sizes [4]. In PCDN, this challenge is compounded by path heterogeneity and device instability.

With the development of Artificial Intelligence (AI), existing work has used Reinforcement Learning (RL) in the preloading control due to its far-sightedness and ability to tackle intricate problems. Wen hu et al. [5] use Deep Reinforcement Learning (DRL) [6] to preload TV series to AP nodes. LiveClip [7] uses DRL to formulate an adaptive short video streaming algorithm. Alfie [8] decides whether to preload the next chunk that is not in the buffer based on RL. With our experience, we find three problems in the existing work: Firstly, most existing research focuses on long videos and fails to consider the unique characteristics of short video scroll events. Secondly, they default to single-path transmission without considering the heterogeneity of multipath. Thirdly, the preloaded size of existing work is inaccurate. In recent years, content providers mostly adopted self-developed transmission protocols. For example, Douyin has developed a stream-pull protocol based on UDP. Therefore, range can be split into finer sizes rather than being limited to the chunk [9]. Fourthly, different videos that a user watches are downloaded from different devices in PCDN, and thus follow different paths. It is difficult to predict the network conditions for preloading the next video while the user is watching the current video, creating additional challenges for preloading.

As one of the world's leading short video services, we make three key observations: firstly, users are more likely to swipe

*Corresponding author: Jiao Zhang

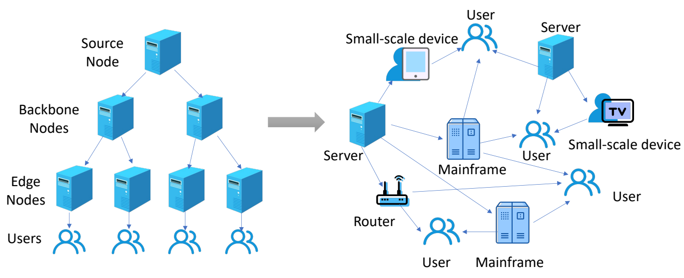


Fig. 1: Transformation of CDN to PCDN transmission architecture.

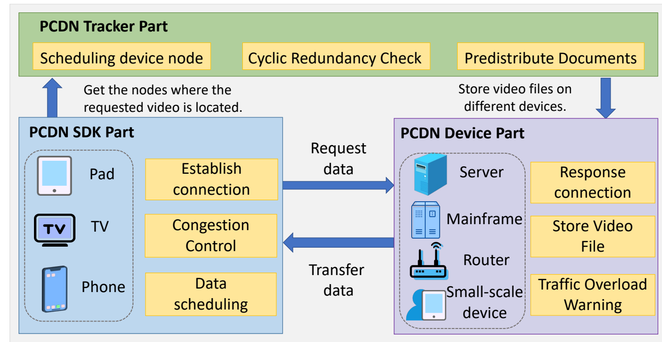


Fig. 2: PCDN Architecture

or exit during watching short videos. Secondly, rebuffering is particularly intolerable for users when it comes to short videos. Thirdly, unnecessary future playback data can result in higher costs than anticipated. Thus, in this paper, we focus on experiences, algorithms, and prospects to solve this cost-QoE dilemma. We present Douyin's PCDN data transmission architecture (§II) and then review learning-based preloading techniques (§III). Farther, we propose a Multipath Smart Preloading algorithm based on RL in PCDN named PreOpt. It decides the range size during playback, aiming to saves costs as much as possible while ensuring QoE (§IV-A). In addition, we design a water level valve algorithm to solve the conflict of preloading multiple video ranges. We notice that changing video bitrates during playback is expensive in PCDN architecture. So we give the video a chance to switch bitrates according to multipath characteristics (§IV-B). Real-world experiments show that our algorithm reduces traffic waste by $\sim$26% while ensuring QoE. Finally, we analyze and outlook the future of multipath video transmission architecture, including trends in PCDN and other open issues, aiming to promote the future development of video architecture.

The main contributions of this paper are:

- We give **a detailed description of Douyin's PCDN** – a multipath short video transmission architecture using fragmented edge resources.
- We propose **a Multipath Smart Preloading algorithm based on RL** in PCDN to solve this cost-QoE dilemma.
- We create **a water level valve algorithm to solve the conflict** of multi-video preloading conflict.
- We analyze and outlook **the future of multipath video transmission architecture**, including trends in PCDN and other open issues.

In what follows, we first introduce the Douyin's PCDN and video preloading in §II. In §IV, we propose a Multipath-aware Smart Preloading algorithm, which consist of three schemes. In §V, we rely on Douyin's PCDN to implement the algorithm and present evaluation results. We analyze and outlook the future of multipath video transmission architecture in §VI.

II. PCDN – A MULTIPATH SHORT VIDEO PLAYBACK ARCHITECTURE

In this section, we first describe the multipath transmission architecture and then analyze its preloading problem.

A. The Multipath Video Transmission Architecture

Short videos, a form of user-generated content shared on online social platforms, are presented to users in a sliding manner [10] and have gained significant success in the mobile video market due to their ability to cater to fragmented user needs and provide large amounts of information in short periods of time. Due to the increasing number of users, content providers are facing significant traffic costs for CDN network transmission of video. As shown in the Figure 1, the key idea of CDN is to distribute the source content to the edge server closest to users and utilize load balancing technology to reduce the response time of user access. The deployment of CDN relies on operators, Internet data centers, service providers, and other factors. While providing stable high-speed bandwidth, CDN comes at a high price.

To reduce cost, Douyin has integrated P2P technology [11] with CDN to build PCDN content distribution network. PCDN exploits the fragmented idle resources of the edge network to build a multipath video distribution system, as illustrated in Figure 1. As depicted in Figure 2, the PCDN system is comprised of three components: the Software Development Kit (SDK), the Device, and the Tracker. The SDK is integrated into the mobile app and is responsible for downloading video playback data for users. When the SDK requires video data, it queries the Tracker for available device nodes from which it can download the video. The SDK then requests the video data in parallel from multiple devices that correspond to the available nodes. *Therefore, each video for a user in PCDN is downloaded from different devices.* The Tracker stores the index mapping from files to devices, and distributes video data to devices according to the popularity of the video. In addition, the Tracker also performs data integrity checks. The Device strives to cover enough videos in advance and transmit the data to the SDK after receiving the request. The Device component comprises idle mainframes or edge access points, which are low-cost but low-bandwidth with a multipath transmission protocol based on UDP. Because it is challenging for a single device node to provide stable and adequate bandwidth to support video playback for a user, PCDN utilizes multiple device nodes to serve each user, thereby enhancing the aggregated bandwidth and ensuring seamless playback.

B. Video Preloading in PCDN Architecture

Due to the characteristics of short videos, users tend to swipe or skip to the next video at any time during playback [4]. To ensure optimal QoE for users, apps preload the current video and videos in the recommendation queue. If the user slides away, the downloaded but unwatched data becomes wasted, increasing the useless overhead of the content provider. As shown in Figure 3, our analysis of data collected from Douyin demonstrates that using a split-range approach (with a fixed 15-second duration for each range) saves approximately ∼12% on average in traffic cost compared to preloading the entire video. Reducing traffic waste while ensuring optimal Quality of Experience (QoE) for users is a challenging task due to conflicting factors. One such factor is the diverse viewing behavior of users, which is influenced by numerous factors that are difficult to model accurately, making it impossible to predict the exact amount of downloads required. Preloading a range that is too small can result in video playback rebuffering when the network quality suddenly deteriorates. Another factor is predicting the download time of data is even more challenging due to the heterogeneity of multipath in the PCDN system. Moreover, different videos that a user watches are downloaded from different devices in PCDN, and thus follow different paths. It is difficult to predict the network conditions for preloading the next video while the user is watching the current video. These factors make it difficult to determine the appropriate order and amount of data to preload.

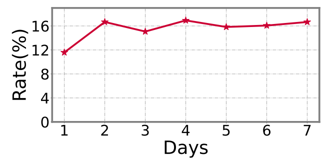


Fig. 3: The daily cost saving ratio after using the split-range download method.

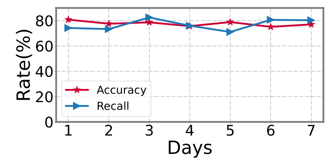


Fig. 4: The recall rate and accuracy rate, which were collected online.

III. Related Work

Recently, with the rapid development of Artificial Intelligence (AI) technology, researchers have started to explore the integration of AI and video playback systems. In this section, we review traditional and learning-based preloading algorithms used in video transmission.

Traditional preloading algorithms: To ensure optimal QoE, content providers such as Douyin have previously preloaded entire videos, resulting in significant waste of traffic costs. Later, a fixed-size range download scheme was derived, where the video is divided into multiple fixed-length ranges and the next range is preloaded when the user starts watching the current range. Another approach is the buffer-based scheme [12], which stops downloading when the video buffer is full. This has a similar effect to the fixed range size scheme. However, traditional preloading algorithms are not flexible enough [13] and often come at the expense of cost in exchange for QoE guarantee, resulting in only limited cost reduction.

Learning-based preloading algorithms: LiveClip [7] uses DRL to develop an adaptive short video streaming policy. It assumes that a video uses a constant bitrate, so each video segment has the same size. It uses the A3C [14] algorithm to decide to download a video segment from 0 to 2 videos in the recommended queue based on the user's current status and download speed. Wen hu et al. [5] use real trajectories to perform large-scale measurements of users' AP connections and TV series watching patterns. They formulate the content preloading problem as a Markov decision process. It uses online learning to preload specific videos to access points to improve user-watching hit rates. Alfie [8] uses the Proximal Policy Optimization (PPO) [15] (A RL algorithm) for deciding whether it is currently necessary to download the next video chunk that is not in the buffer, based on the current behavior of the user and the network throughput. The n videos in Alfie may have different preloading progress. Incendio [16] is an innovative SABR (short video adaptive bitrate) framework that utilizes Multi-Agent Reinforcement Learning (MARL) with Expert Guidance. It separates the decision-making process for video ID and video bitrate into buffer management and bitrate adaptation agents, respectively.

In summary, the above learning-based preloading algorithms are not suitable for the preloading control problem in PCDN. Firstly, they assume that all videos in a user session are downloaded from the same CDN, and therefore use network parameters obtained from past downloads when preloading different videos in the queue. However, in PCDN, each video may be located on different devices, and the network parameters of the current video cannot represent any other video. Additionally, maintaining connection states for multiple videos, as in the aforementioned algorithms, would incur significant overhead for mobile devices, given that each video in PCDN transfers data from multiple devices in parallel. Therefore, implementing a preloading control algorithm that achieves a balance between QoE and cost, perfectly adapted to the PCDN architecture, requires addressing the following challenges: 1) How to anticipate the transmission path status of the recommended video queue in advance? 2) How to integrate the network parameters of multiple paths into the network state space of RL? 3) How to predict the user's sliding probability?

Given the above complications, we use DRL to solve the preloading problem in PCDN multipath video transmission. DRL can determine the next range size based on empirical information through a long-term optimization approach. So the algorithm can combine many factors to decide on a more accurate result. Additionally, the paper addresses the characteristics of short video sliding and the PCDN transmission architecture to predict network state information and designs the Water Valve Algorithm to solve range download conflicts. Details in §IV.

IV. Design

Figure 5 depicts the training and video downloading process. It consists primarily of three components, and the following sections will provide detailed descriptions of these three modules.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

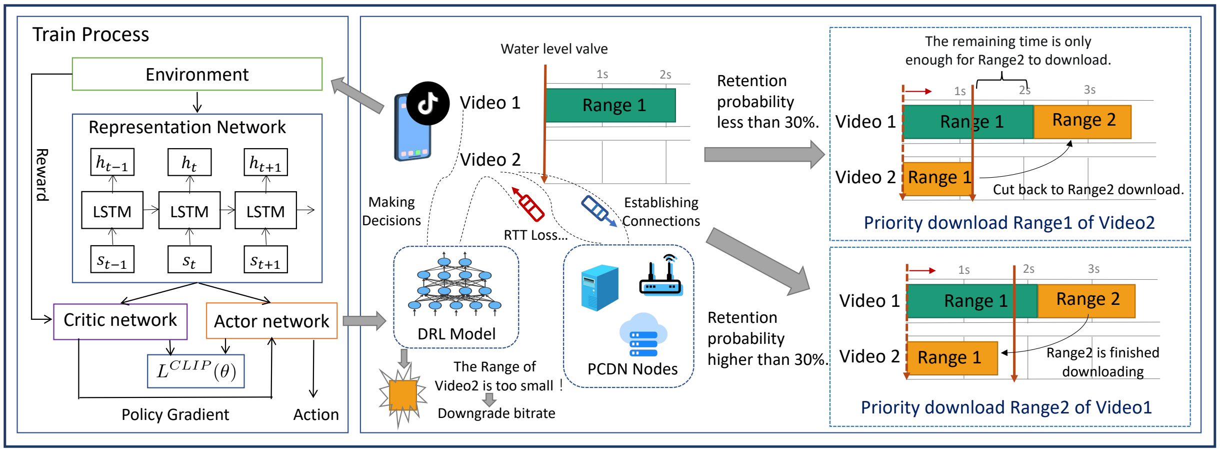


Fig. 5: Design overview: the training process and the video downloading process

A. Model Formalization

1) State: The state is the environmental information that is input to the agent. We divide the input information into three parts based on the factors influencing the range size decision: network state, user behavior, and playback state.

- Network state: Network fluctuations affect whether the next range will rebuffer. We denote the network state as $N_t = [b_t, r_t, l_t]$, which are the throughput, RTT, and loss rate at time t, respectively. Where throughput is the download speed of the multi-transmission task aggregated from all the connections, the loss rate is the number of packets lost by the task divided by the number of packets sent by the task, and RTT is weighted together by the throughput of each connection. The network state is combined the conditions of all the links.
- User behavior: Another key factor influencing range size is user-watching behavior. We predict the probability of user swiping by $U_t = [o_t, d_t]$. o_t is the recommendation score of the video given by the recommendation algorithm, which is used to represent the user's level of interest in the current video. d_t is the historical swiping rate of the user, which has been accumulated statistically.
- Playback state: The playback information input $P_t = [e_t, m_t]$ includes the bitrate e_t of the video and the remaining supportable watching time m_t of the current range at time t.

2) Action: The action at epoch t is the length of time supported by range, in seconds.

3) Reward: For short videos, the main factors affecting QoE are clarity and rebuffering. The cost is the traffic spend. To sum up, the reward of $range_i$ is $R_t = w_1 \cdot bitrate_i - w_2 \cdot rebuffer_i - w_3 \cdot range_{i-1}\_remain - w_4 \cdot traffic\_usage$. $range_{i-1}\_remain$ is the undownloaded part of the previous range. At the beginning of a new range, we lack information about the rebuffering situation. Therefore, we use the part of the current range that has not been downloaded to replace it that tends to download the next range in the previous range to ensure smooth playback. Additionally, we use the duration of the previous range as a penalty item in the reward function to prevent the algorithm from making extreme decisions.

4) Learning Algorithm: On the left side of Figure 5, we utilize the Proximal Policy Optimization (PPO) algorithm to train the policy. PPO is a gradient descent algorithm based on the Actor-Critic network, which uniquely employs the $CLIP$ function to prevent policy fluctuations caused by large learning steps. In the preloading problem, the user's previous behavior provides valuable information for determining the current range size. However, conventional neural networks lack memory functions. To address this, we train the input using the Long Short-Term Memory (LSTM) Recurrent Neural Network algorithm as a representative network.

During video playback, the DRL model is utilized to determine and preload the size of the next range whenever the user begins watching a range.

B. Water Valve Algorithm resolves conflicts in cross-video preloading.

Because short video playback involves sliding, preloading control must consider preloading videos from the recommendation queue. However, preloading multiple videos from the queue can result in maintaining too many connection states on mobile devices in PCDN, leading to higher energy consumption and costs (maintaining 20 links states results in a 2.5% increase in battery drain). We have observed that during short video playback, users can only slide through videos in the order presented in the recommendation queue and will pause on the next video for a few seconds to decide whether to continue watching or not. Therefore, we simplify the problem of preloading videos from the recommendation queue by preloading the first range of the next video at the beginning of the current video, ensuring the smooth playback of videos. However, downloading two ranges at the same time can confuse the DRL model. In the following sections, we will describe in detail the problem of range download conflicts and our proposed solutions.

Cross-video Range Download Conflict: As mentioned earlier, preloading the first range of two videos is necessary at the beginning of a video. However, downloading two ranges at the same time can lead to competition for the user's bandwidth. The network state parameters used by the DRL model to

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

determine range size in previous decisions being incorrect can cause both ranges to fail to download successfully, leading to conflicts.

Water valve algorithm: We conceptualize the user's watching position as a water valve, and our algorithm makes corresponding decisions based on its movement. As shown in Figure 5, $Range_1$ already exists due to the preloading of the previous video when the user starts watching $Video_1$. If $Video_1$'s $Range_1$ has not finished downloading at this point, we prioritize finishing its download. Meanwhile, our model determines the size of $Video_1$'s $Range_2$ and $Video_2$'s $Range_1$, and the algorithm chooses how to download these two video ranges.

We use the retention probability generated by the recommendation algorithm to choose the next action. Fig. 4 shows that Douyin's retention probability prediction has high accuracy and recall rates, making it reliable. If the user's retention probability is less than a threshold η, algorithm prioritizes downloading $Video_2$'s $Range_1$ as the next range. At the same time, the water valve moves backward as the user watches. When the water level valve reaches the point where the remaining watching time of $Video_1$'s $Range_1$ is equal to the download time of $Video_1$'s $Range_2$, and the user has not jumped to $Video_2$ yet, we jump back to download $Video_1$'s $Range_2$. Otherwise, we download $Video_1$'s $Range_2$ when $Video_2$'s $Range_1$ has finished downloading. We calculate the download time of a range by dividing the total download rate by its size and the smooth RTT. If the user's retention probability is higher than η, we first download $Video_1$'s $Range_2$ and then download $Video_2$'s $Range_1$. If the user has already watched $Video_1$'s $Range_2$ and $Video_2$'s $Range_1$ has not finished downloading, we follow the current pattern for the next range.

Establishing connections to obtain path information: One challenge with the water valve algorithm is the need for path information for $Video_2$ when deciding its $Range_1$. In a PCDN system, different files are likely to be stored on separate devices, resulting in distinct network states for $Video_2$ compared to those obtained for $Video_1$ (§III). However, it is worth noting that in the PCDN system, the use of multipath is intended to increase the aggregated bandwidth, with the bottleneck most likely to be on the user side. To address this challenge, we establish connections with $Video_2$'s devices before model decides the range size to obtain information, such as RTT. We then combine this data with the throughput derived from $Video_1$ to determine the appropriate size for $Video_2$'s $Range_1$. By taking this approach, we can make informed decisions about range size and ensure that the PCDN system efficiently utilizes its network resources.

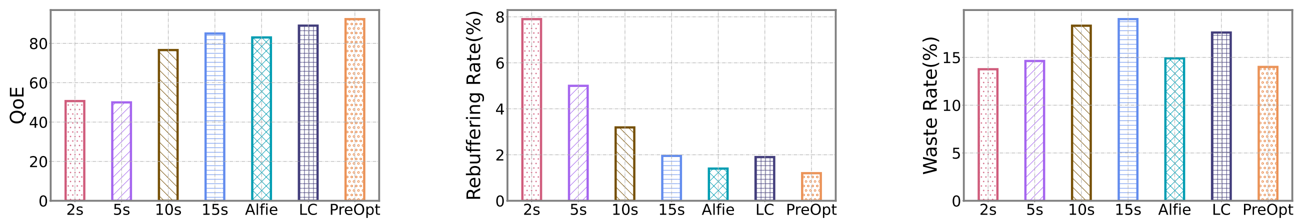
C. Video bitrate adjustment

In the PCDN system, a video determines its bitrate at the beginning of the transmission. Given that different bitrate files of a video may be stored on different device nodes, modifying the bitrate requires re-querying the nodes and establishing new connections, which results in additional time and resource overheads. As such, the video only connects with devices once the bitrate is determined. The PCDN determines the video bitrate based on its speed. However, in the above algorithm, the bitrate of $Video_2$ can only be decided based on the speed of $Video_1$. To reduce the impact on subsequent video playback, we have designed a range comparison algorithm that allows the video to change its bitrate. When $Video_2$ is connected, we obtain new path information to determine the size of $Video_2$'s $Range_1$. If the size of $Range_1$ is less than the threshold (3s duration), it indicates that the network is unable to support the current bitrate. In such cases, we adjust the bitrate downward by one level if the watchable time of the range is between 1 and 3 seconds. If the watchable time of the range is less than 1 second, we adjust the bitrate downward by two levels. This algorithm enables videos to adjust their bitrates dynamically, thereby providing users with a seamless and uninterrupted viewing experience.

V. EVALUATION

Evaluation setting: We rely on Douyin's PCDN to implement PreOpt and collect user data. Approximately 100,000 Douyin users updated PreOpt and participated in an online experiment, generating millions of video data. Before using the model, we trained the agent over 1000000 epochs under different scenarios, which took about 24 hours. And the model has been refined among real users online for nearly three months. The retention probability threshold (η) is set at 30%. Based on online experience, this value has proven to strike a good balance. We also included two representative works, LiveClip (LC) and AIfile, which use RL-based preloading, in the experiment. A fixed range preloading strategy was used as a comparison. In the results, the values of 2s, 5s, 10s, and 15s represent the size of the fixed range.

Results: We used the approach in [13] to calculate QoE, which involves subtracting the rebuffering rate from the bitrate in our paper. Fig. 6a displays the average QoE for each video in the online experiment. Compared to other strategies, PreOpt improved QoE by 3%$\sim$80%. Figure 6b shows the rebuffering rate, a key performance indicator for short videos. Fig. 6c displays the average bandwidth waste rate for each video. We observed that smaller range sizes result in higher rebuffer rates. If the range is too small, rebuffering time will be too long, and video playback will fail. For example, the strategy with a range size of 2s has the lowest waste rate but high buffering rate. However, if the range size is too large, more traffic will be wasted. The RL-based algorithm we designed balances these two aspects well because the RL-agent can take multiple factors into consideration. PreOpt reduced buffering rate by 14%$\sim$80% while reducing waste rate by 2%$\sim$26%. In addition, prior to the final presentation of the results, each component underwent its own evolutionary process. When the Water Valve algorithm was not present, relying solely on DRL for range size decision led to a reduction of approximately 20% in waste rate while maintaining a lower buffering ratio. The introduction of the Water Valve algorithm further decreased the buffering ratio by enabling preloading of the next video, resulting in optimal QoE maintenance at a threshold of 30%. Setting the threshold too high or too low may cause



(a) PreOpt brings higher QoE.

(b) Preopt effectively reduces the rebuffering rate.

(c) PreOpt improves QoE while reducing waste.

Fig. 6: Evaluation Results.

rebuffering in the current or next video. Based on our online experience, we find that setting the threshold to 30% yields the best results.

VI. Future Work and Conclusion

Multipath video transmission is still in an embryonic stage and there is huge potential for development. The huge cost saving and resource utilization of multipath video transmission has attracted wide attention from the industry. However, there are still many points worth research in the PCDN architecture in addition to the adjustment of range size. The first is the file distribution mechanism. Since multiple small devices are used to store files in PCDN, it is very important to allocate the file placement. According to the popularity, location and other information, the reasonable allocation of files is advantageous to the saving of storage space. The second is the problem of selecting the devices to download video. The same video may be stored in many devices. A poor connection may affect the download of the entire file. The third is the data scheduling. Due to the heterogeneity of the paths there will be data disorder, thus causing queue head blocking. In conclusion, there are still many issues worthy of optimization in the PCDN architecture, which still need to be researched in the future.

In this paper, we introduce PCDN, a multipath video transmission architecture that leverages edge residual resources to store and transfer videos, thereby saving content providers significant traffic costs. We propose a range size adjustment algorithm based on DRL in the PCDN architecture to enhance video watching quality and user experience while reduce cost. The algorithm uses the user's viewing position as a water level valve and makes corresponding decisions based on its movement. It also considers video preloading and bitrate adjustment to optimize video download and playback efficiency. Our algorithm achieves good performance in experiments and provides useful references for future research and development in the field of video streaming transmission technology and systems.

Acknowledgments

This work is partly supported by the National Key R&D Program of China under Grant No. 2022YFB2901700, the National Natural Science Foundation of China (NSFC) under Grant No. 62132022, Fok Ying Tung Education Foundation under Grant No. 171059, and BUPT Excellent Ph.D. Students Foundation (No. CX2021102)

References

[1] L. Zhang, F. Wang, and J. Liu, "Understand instant video clip sharing on mobile platforms: Twitter's vine as a case study," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, 2014.
[2] G. Peng, "Cdn: Content distribution network," *arXiv preprint cs/0411069*, 2004.
[3] X. Wu, L. Zhang, and L. Cui, "Qoe-aware download control and bitrate adaptation for short video streaming," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.
[4] Z. Chen, Q. He, Z. Mao, H.-M. Chung, and S. Maharjan, "A study on the characteristics of douyin short videos and implications for edge caching," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019.
[5] W. Hu, Y. Jin, Y. Wen, Z. Wang, and L. Sun, "Toward wi-fi ap-assisted content prefetching for an on-demand tv series: A learning-based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
[6] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017.
[7] J. He, M. Hu, Y. Zhou, and D. Wu, "Liveclip: towards intelligent mobile short-form video streaming with deep reinforcement learning," in *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2020.
[8] J. Li, H. Xu, M. Ma, H. Yan, and C. J. Xue, "Alfie: Neural-reinforced adaptive prefetching for short videos," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022.
[9] R. Pereira and E. G. Pereira, "Dynamic adaptive streaming over http and progressive download: Comparative considerations," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2014.
[10] W. Hu and G. Cao, "Energy-aware video streaming on smartphones," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015.
[11] J.-F. Paris and P. Shah, "Peer-to-peer multimedia streaming using bittorrent," in *2007 IEEE International Performance, Computing, and Communications Conference*. IEEE, 2007.
[12] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014.
[13] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015.
[14] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016.
[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[16] Y. Li, Q. Zheng, Z. Zhang, H. Chen, and Z. Ma, "Improving abr performance for short video streaming using multi-agent reinforcement learning with expert guidance," in *Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video*, 2023, pp. 58–64.

VII. Biography Section

Dehui Wei received the B.S. degree in computer science and technology from Hunan University, Changsha, China, in 2019 and was awarded outstanding graduate. She is currently working toward her Ph.D. degree at the State Key Laboratory of Networking and Switching Technology at Beijing University of Posts and Telecommunications (BUPT). Her research interests are in the areas of network transmission control and cloud computing.

Jiao Zhang is currently an associate professor in the School of Information and Communication Engineering at Beijing University of Posts and Telecommunications, China. In July 2014, she received her Ph.D degree from the Department of Computer Science and Technology, Tsinghua University, China. From Aug. 2012 to Aug. 2013, she was a visiting student in the networking group of ICSI, UC Berkeley. In July 2008, she obtained her Bachelor Degree from the School of Computer Science and Technology from BUPT. Her research interests mainly include data center networking, network function virtualization, and future Internet.

Haozhe Li received a B.S. degree in communication engineering from Wuhan University of Technology, Wuhan, China, in 2020. He is currently working toward his M.S. degree at the State Key Laboratory of Networking and Switching Technology at Beijing University of Post and Telecommunications(BUPT). His research interests are in areas of data center networks and network congestion control.

Zhichen Xue received his M.S. degree in Computer Science from the Illinois Institute of Technology, Chicago, IL, USA, in 2021. After graduation, he joined Bytedance Ltd. Beijing, China, as a software development engineer. His current interests include high-performance network programming, multi-path transport, and network simulation.

Yajie Peng received her M.S. degree in communication engineering from Huazhong University of Science and Technology, Wuhan, China, in 2015. She is currently a Staff Engineer at ByteDance Inc. Her work focus on multipath transmission optimization for VoD and live-streaming.

Rui Han received his M.S. degree from the University of Hong Kong in 2013. He is currently a Staff Engineer and the Leader of Hyper Edge Nodes Team at ByteDance Inc. His work focus on distributed network, multimedia, and cloud computing. Currently he builds PCDN, the untraditional cost-effective CDN system running on very large-scale, heterogeneous, and hyper edge devices on the Internet.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

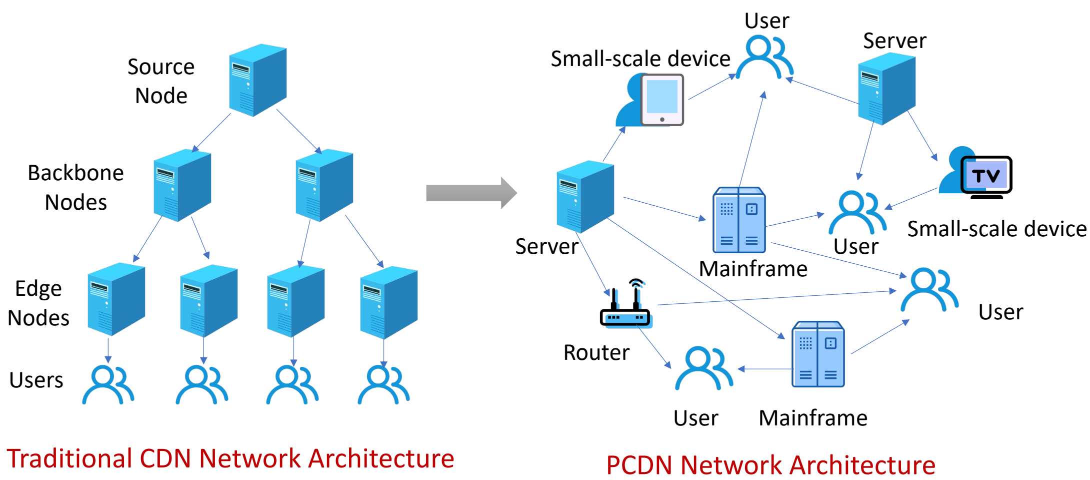


Fig. 1: Transformation of CDN to PCDN transmission architecture.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

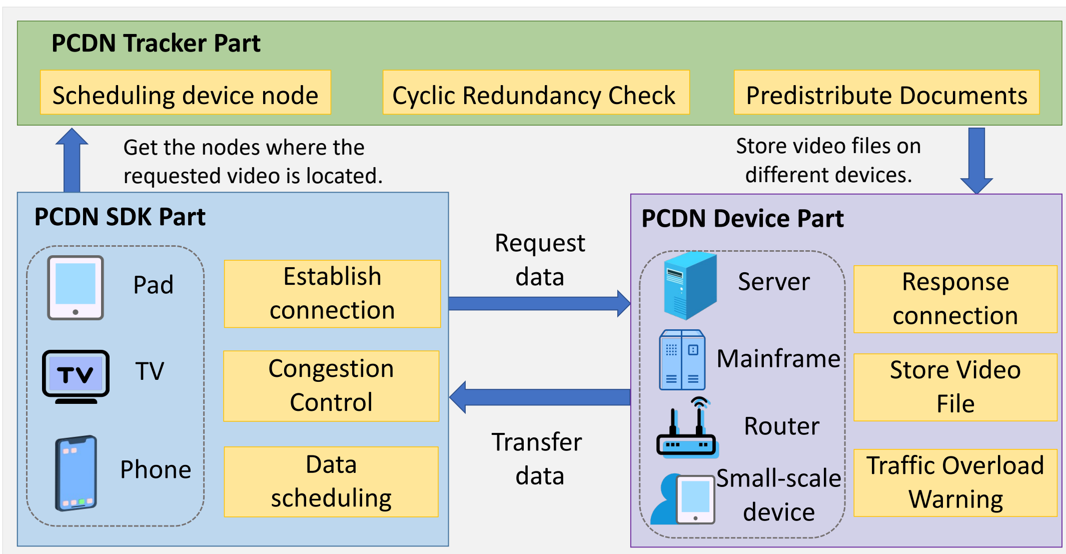


Fig. 2: PCDN Architecture

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

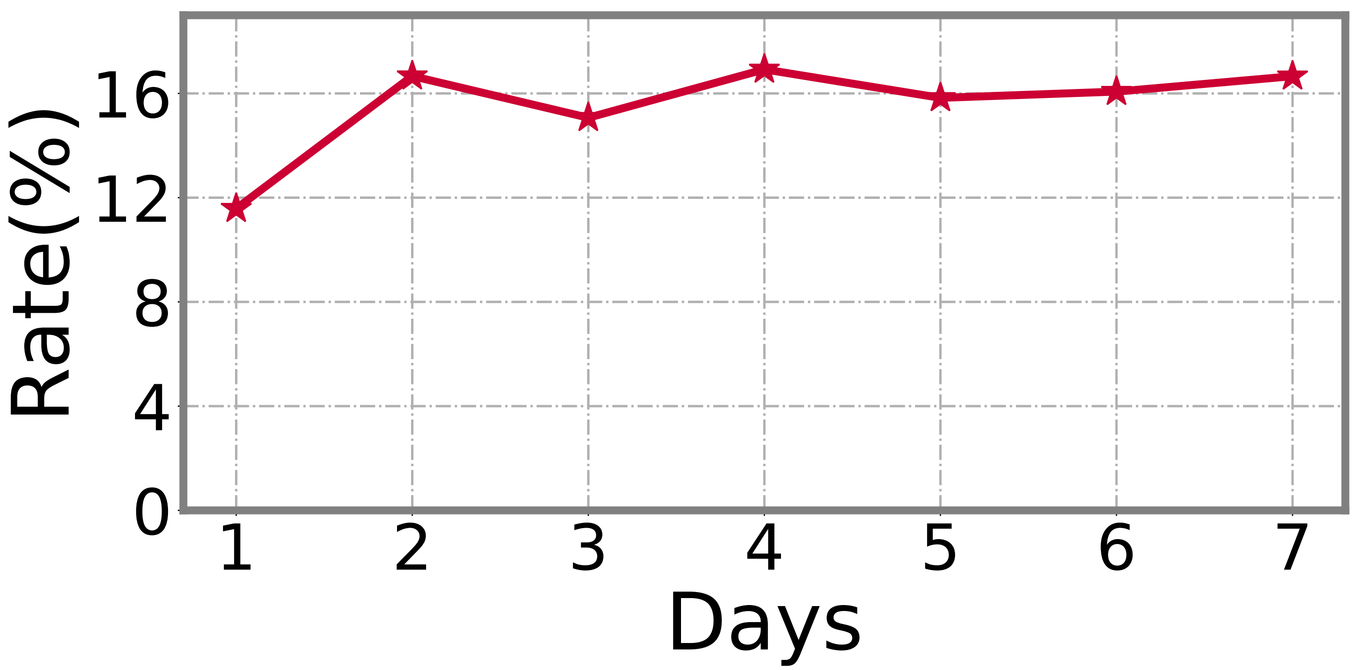


Fig. 3: The daily cost saving ratio after using the split-range download method.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

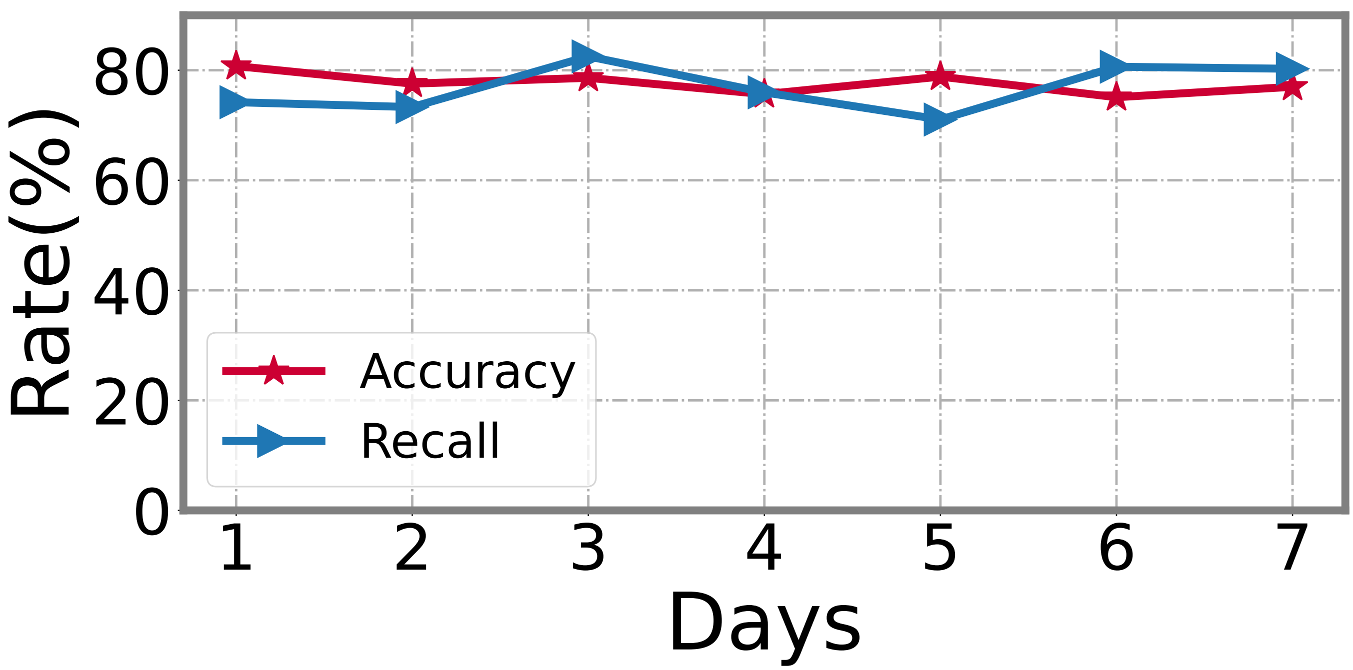


Fig. 4: The recall rate and accuracy rate, which were collected online.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526

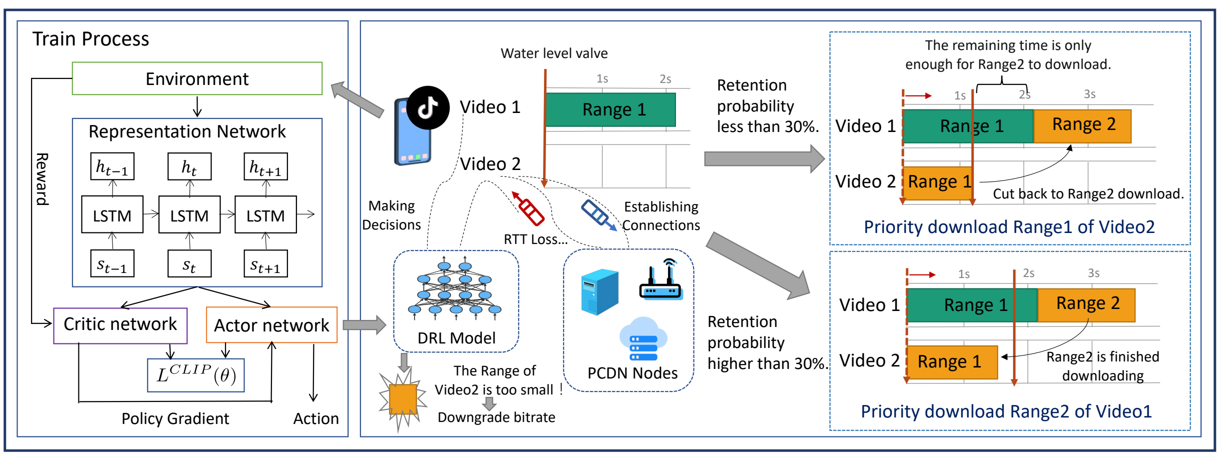
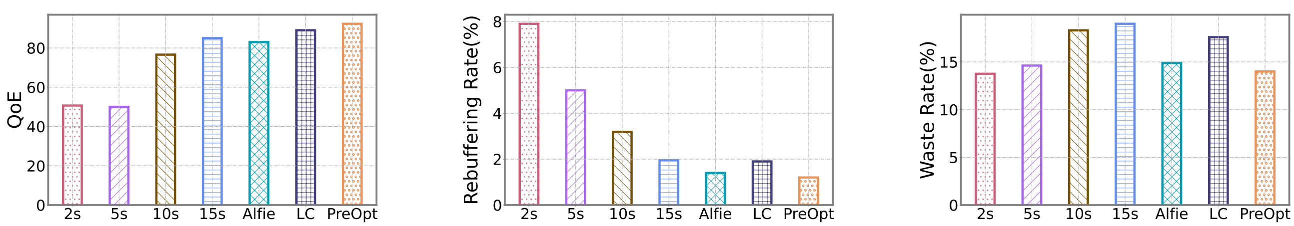


Fig. 5: Design overview: the training process and the video downloading process

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MNET.2023.3321526



(a) PreOpt brings higher QoE.

(b) Preopt effectively reduces the rebuffering rate.

(c) PreOpt improves QoE while reducing waste.

Fig. 6: Evaluation Results.