QoE-driven Joint Decision-Making for Multipath Adaptive Video Streaming

Jinwei Zhao, Jianping Pan
Department of Computer Science, University of Victoria
{clarkzjw, pan}@uvic.ca

***Abstract*—Multipath transport protocols including multipath TCP (MPTCP) and multipath QUIC (MPQUIC) are designed to utilize multiple network paths for simultaneous data transfer. These protocols try to improve network performance and offer better resilience in dynamic network environments. Nonetheless, the actual performance improvement is heavily reliant on the effectiveness of the multipath scheduling algorithms. In specific scenarios such as adaptive video streaming, most existing solutions feature two separate and independent control loops for multipath scheduling and video bitrate adaptation, while multipath scheduling algorithms are usually transparent to the video bitrate adaptation process. Lacking the context of inter-path differences and intra-path fluctuations for both network throughput and latency may potentially result in a suboptimal quality of experience (QoE) for video streaming. Such circumstances may lead to a reduced video bitrate, increased latency, and a greater number of rebuffering events. In this paper, we present a QoE-driven joint decision-making framework based on contextual multi-armed bandit (CMAB) algorithms to efficiently address multipath adaptive video streaming problems. This approach merges application-layer (playback buffer ratio) and network-layer (throughput and latency) metrics to create a context-aware online learning model, which can adaptively select the ideal network path and bitrate for multipath adaptive video streaming. Both network emulation and real-world experiments demonstrate that the proposed algorithm delivers better QoE, including higher average video bitrate and fewer rebuffering events when compared to independent decision-making algorithms.**

***Index Terms*—Multipath QUIC, adaptive video streaming, multi-armed bandit**

I. Introduction

Video streaming currently dominates Internet traffic and is expected to continue growing in the future. Cisco estimated that by 2023, 66% of smart TV installations will be 4K UHD, up from 33% in 2018 [1]. For video streaming applications, users' quality of experience (QoE) is significantly influenced by network latency, perceived video quality, and the number of bitrate fluctuations. Due to the heterogeneous and dynamic nature of the Internet, delivering high-quality video-on-demand (VoD) services remains a formidable challenge.

Throughout the years, numerous adaptive video streaming methodologies and algorithms have been proposed to enhance the QoE. Dynamic Adaptive Streaming over HTTP (DASH) [20] has become the international standard for adaptive video streaming. Previously, HTTP was built with the underlying transport protocol TCP. TCP suffers from high latency connection establishment and head-of-line (HoL) blocking issues, making it unsuitable for latency-critical video streaming scenarios such as live streaming and video conferencing on the modern Internet.

QUIC is a new UDP-based transport protocol to enhance the performance and security of Internet connections. It has recently been proposed as RFC 9000 [11] by IETF and forms the foundation of HTTP/3 [5]. QUIC's built-in features, such as stream multiplexing, resolve the HoL blocking issue. Additionally, 1-RTT connection establishment and 0-RTT connection resumption considerably reduce network latency, making QUIC a promising option for scenarios that require fast and reliable communications. Unlike the TCP stack, which is implemented in the kernel, QUIC is based on UDP and designed for implementation in the user space. This approach allows for easier integration with existing applications and network infrastructures without creating protocol ossification issues.

In recent years, multipath transport protocols have attracted increased attention [17], [8]. Multipath TCP (MPTCP) [9] was proposed as an extension to TCP, enabling the utilization of several distinct network paths for concurrent data transmission to enhance both network reliability and performance. MPTCP faces challenges in efficiently developing new scheduling algorithms and achieving widespread adoption due to the high optimization of the existing TCP network stack. On the other hand, multipath QUIC (MPQUIC), one of the ongoing QUIC extension drafts, has collected significant interest in the community [18]. Similar to MPTCP, MPQUIC aims to utilize multiple network paths for simultaneous data transmission to increase reliability and performance. Additionally, it also leverages QUIC's advantages such as stream multiplexing, 1-RTT connection establishment, and 0-RTT connection resumption. MPQUIC presents a new opportunity to develop state-of-the-art multipath scheduling algorithms in the user space, integrating application-specific contexts with less effort.

The majority of current multipath adaptive video streaming solutions employ two separate and independent algorithms for multipath scheduling and video bitrate adaptation [25], [27]. At the transport layer, network metrics such as throughput and latency are typically employed to carry out the multipath packet scheduling process. However, at the application layer, video players lack the context regarding transport layer packet scheduling algorithms. Consequently, the inter-path differences and intra-path fluctuations of network throughput and latency can impact the QoE of various video streaming scenarios. However, reconciling both control loops with empirical rules can be challenging due to their distinct objectives.

In short, the main contributions of our work can be summarized as follows:

- First, we model the multipath adaptive video streaming problem as an online learning process. Contextual multi-armed bandit (CMAB) algorithms are employed to solve the online learning problem, as they offer satisfactory results and are significantly more lightweight than deep reinforcement learning-based techniques.
- Second, we design a QoE-driven reward function for the CMAB problem formulation to sustain a high video bitrate while reducing rebuffering events.
- Finally, the proposed algorithm is extensively evaluated with both network emulation and real-world multipath testbed experiments. Both results demonstrate the significant potential of our joint decision-making framework in improving the QoE of multipath adaptive video streaming.

The rest of the paper is organized as follows. Section II introduces the related works on adaptive video streaming and multipath transport protocols. Section III presents the system model and problem formulation of the proposed multipath adaptive video streaming algorithm. Section IV illustrates the extensive performance evaluation with Mininet emulation and real-world testbed experiments. Section V concludes the paper while discussing the remaining challenges and potential future work.

II. Related Works

In this section, we provide a brief overview of the state-of-the-art algorithms in both adaptive video streaming and multipath transport protocols.

A. Adaptive Video Streaming

In this paper, we classify existing adaptive bitrate streaming (ABR) algorithms into two categories based on a widely accepted criterion [13]: (1) client-side techniques including throughput-based, buffer-based and hybrid algorithms; and (2) server-side or SDN-assisted techniques.

Traditional client-side ABR algorithms rely on network measurements including available bandwidth estimation or application contexts such as local playback buffer ratio. Spiteri et al. [19] proposed the buffer-based *BOLA* algorithm, which formulates bitrate adaptation as a utility-maximization problem. Lyapunov optimization is utilized to minimize rebuffering and maximize video quality. *BOLA* and its variants have been integrated into the DASH reference player *dash.js* [2] and are widely used in production. However, bitrate adaptation algorithms based on fixed sets of empirical rules often suffer from low performance in severely dynamic and unpredictable network environments. To address this issue, Cui et al. [6] used deep reinforcement learning to achieve the joint optimization of video bitrate, rebuffering time and latency. However, most deep reinforcement learning-based ABR algorithms are computationally intensive and require powerful hardware such as GPUs. As a result, they are typically deployed on the server side. Yeo et al. [26] proposed *NEMO*, which uses server-assisted techniques to enable real-time video super-resolution on mobile devices to improve the QoE of adaptive video streaming. On the other hand, SDN-assisted ABR algorithms are promising in theory but usually lack real-world end-to-end support. Farahani et al. [7] proposed *ES-HAS* to enable adaptive video streaming over SDN-enabled networks. Virtualized edge components and an SDN controller are deployed to collect streaming clients' requests and retrieve underlying network information to determine the optimal cache server for clients. Nevertheless, provided that end-to-end SDN infrastructure cannot be guaranteed between the video streaming client and the server, the advantages of SDN-assisted algorithms vanish and the video streaming experience fallbacks to whatever the ABR algorithm can provide.

B. Multipath Transport Protocols

With proper configuration, MPTCP is expected to provide better network performance, such as higher throughput [12], [22], increased network resilience and redundancy in the face of network failures [15]. Xing et al. [24] proposed *OLAPS* for MPTCP to generate adaptive scheduling policies and quickly respond to abrupt network changes. *OLAPS* uses UCB1 with a custom reward function based on instantaneous throughput at both the subflow and connection levels, rather than relying on network measurements of congestion windows, RTT, and loss rates. Zhao et al. [27] proposed *MPTCP+*, which incorporated the path use decision and the multipath congestion control, to address the problems of inter-path throughput difference and congestion window fluctuation. On the other hand, while the MPQUIC extension draft is still under IETF discussion [18], many efforts have already been made to explore different scheduling algorithms for it. *Peekaboo* is a learning-based MPQUIC scheduler presented by Wu et al. [23]. *Peekaboo* utilized LinUCB [16] to address the multipath scheduling problem in heterogeneous network environments with dynamically changing path characteristics.

III. System Model and Problem Formulation

Typically, the system model of a multipath adaptive video streaming problem consists of two independent control loops, the ranking and selection process for different network paths and the adaptation for video bitrates. The multipath scheduling process is usually transparent to the video bitrate adaptation. In this work, we focus on building a joint decision-making framework employing CMAB algorithms to address both issues simultaneously. The notations used in this paper are summarized in Table I.

A. Adaptive Video Streaming

DASH [20] workflow begins with the source video being encoded into various bitrate levels with a bitrate ladder and then segmented into smaller chunks usually with a fixed duration (e.g., 2 seconds). The video segments along with a media presentation description (MPD) file containing necessary metadata about the available bitrates and segment URLs

TABLE I: Summary of Key Notations

Notation	Definition
K	Number of arms
T	Total number of rounds
$a(t)$	Arm selected by agent at round t
$b(t)$	Context vector revealed to agent at round t
μ_k	Distribution parameter for path k
$\mathcal{H}_{t-1}$	History up to round $t-1$
$a^*(t)$	Optimal arm at round t
N	Total number of video segments
P	Total number of network paths
M	Total number of available bitrate levels
C	Total number of rebuffering events
$\mathrm{R}(t)$	Playback buffer ratio at time t
$\mathrm{RTT}_p(t)$	Estimated average RTT for path p at time t
$\mathrm{BW}_p(t)$	Estimated average throughput for path p at time t
$r_{(m,p)}(i)$	Reward of arm (m,p) for video segment i
t_j	Time duration for rebuffering event j
RBF_t	Rebuffering time ratio at time t
$\mathrm{B}(i)$	Bitrate of video segment i
$\mathrm{B}_{\max}$	Highest video bitrate available in the MPD file

for streaming clients are then stored on a web server. When a client initiates a video streaming session, the client retrieves the MPD file and starts downloading the video segments based on the current network conditions, device capabilities, and user preferences. As the streaming session progresses, the client continuously monitors these factors and adjusts the bitrate level of the video segments to be downloaded. By dynamically switching between different bitrate levels, DASH ensures that the client can receive the best possible video quality under dynamic network conditions while minimizing the risk of rebuffering events, resulting in a seamless playback experience.

B. Problem Formulation

The contextual multi-armed bandit problem can be defined as follows [4]. There are K arms. At each round $t = 1, 2, \ldots, T$, a context vector $b(t) \in \mathbb{R}^d$ is revealed to the agent and the agent pulls the arm which can yield the highest expected reward. The agent is only informed of the reward for the pulled arm. Rewards for the unselected arms remain unknown in each round. After observing the arms played and their rewards up to round $t-1$, the history $\mathcal{H}_{t-1}$ can be defined as,

$$\mathcal{H}_{t-1} = \left\{a(s), r_{a(s)}(s), b(s), s = 1, \ldots, t-1\right\} \tag{1}$$

where $a(s)$ denotes the arm played at round s and $r_{a(s)}(s)$ is the reward for arm $a(s)$ at round s. Assume given $b(t)$, the reward for arm k at round t is generated from an unknown distribution with mean $b(t)^T\mu_k$, where $\mu_k \in \mathbb{R}^d$ is a fixed but unknown parameter to the agent. $b(t)^T$ denotes the matrix transpose of $b(t)$. The expected reward of $r_k(t)$ for each arm k given $b(t)$ and $\mathcal{H}_{t-1}$ can be defined as,

$$\mathbb{E}\left[r_k(t) \mid b(t), \mathcal{H}_{t-1}\right] = b(t)^T\mu_k. \tag{2}$$

In the setting of a contextual multi-armed bandit problem, an agent with an online algorithm needs to choose, at every round t, an arm $a(t)$ to pull based on the history $\mathcal{H}_{t-1}$ and the context of the current round $b(t)$ revealed to the agent. Let $a^*(t)$ denote the optimal arm at time t, which yields the highest expected reward, i.e., $a^*(t) = \arg\max_k b(t)^T\mu_k$. Let $\Delta_k(t)$ be the difference between the reward of the optimal arm $a^*(t)$ and arm k at time t, i.e.,

$$\Delta_k(t) = b(t)^T\mu_{a^*(t)} - b(t)^T\mu_k \tag{3}$$

Then, the regret at time t is defined as

$$\mathrm{regret}(t) = \Delta_{a(t)}(t) \tag{4}$$

It is important to note that the CMAB problem setting presented here deviates slightly from the one in [4]. In our case, we assume that each arm k is revealed with the same context vector $b(t)$, and each arm adheres to an unknown yet distinct distribution characterized by different μ_k.

In this paper, the multipath adaptive video streaming problem is formulated with CMAB settings as follows. There are N video segments and P available network paths. Each video segment is encoded into M different bitrate levels. The CMAB action space is defined as the cartesian product $M \times P = \{(m,p) | m \in M \,\text{and}\, p \in P\}$. An arm is defined as any possible combination of available video bitrate and network path (m,p).

We define the context vector $b(t)$ as follows,

$$b(t) = \{\mathrm{R}(t), \mathrm{RTT}_p(t), \mathrm{BW}_p(t), p = 1, 2, \ldots, P\} \tag{5}$$

where at time t, $\mathrm{R}(t)$ is the local playback buffer ratio, $\mathrm{R}(t) \in [0,1]$. $\mathrm{RTT}_p(t)$, $\mathrm{BW}_p(t)$ is the estimated average RTT and throughput for path p at time t respectively. Typically, a low playback buffer ratio $\mathrm{R}(t)$ indicates that the network cannot sustain the required video bitrate. Thus, a rebuffering event would occur unless the video streaming client adjusts to a lower video bitrate. However, this assumption does not always hold. A high playback buffer ratio does not necessarily mean the network is healthy because the player could be just downloading many low-bitrate video segments. Thus, the local playback buffer ratio alone cannot be referred to as a reliable context for the player to make video bitrate and multipath scheduling decisions. Therefore, $\mathrm{RTT}_p(t)$ and $\mathrm{BW}_p(t)$ are added to the context vector $b(t)$ as additional information to help the player make reliable decisions.

We develop a QoE-driven reward function with the primary objective of minimizing rebuffering events and increasing playback bitrate. In VoD scenarios, users' QoE is usually less sensitive to network latency, compared with live video streaming, and largely affected by video bitrate and rebuffering events. We define the rebuffering time ratio at time t as follows,

$$\mathrm{RBF}_t = \frac{\sum_{j=1}^{C} t_j}{t}, \tag{6}$$

where C is the total number of rebuffering events that happened up to time t and t_j is the time duration for each rebuffering event j. The agent pulls an arm before downloading each video segment i, and the corresponding reward $r_{(m,p)}(i)$ for video segment i obtained from path p is defined as,

$$r_{(m,p)}(i) = -1 * \mathbf{1}\{\mathrm{RBF}_t - \mathrm{RBF}_{t-1} > 0\} * \frac{\mathrm{B}(i)}{\mathrm{B}_{\max}} \quad (7)$$

where $\mathbf{1}$ is an indicator function of $\{-1, 1\}$. $\mathrm{B}(i)$ is the video bitrate for video segment i and $\mathrm{B}_{\max}$ is the maximum bitrate available in the MPD file. When $\mathrm{RBF}_t - \mathrm{RBF}_{t-1} > 0$, it indicates that the current selection of bitrate and path introduces interruptions to the playback. Thus, a negative reward is revealed to the agent. The objective of the agent is to pull the best arm which yields the highest expected reward in each round for video segment i, i.e., the video player decides the best combination of network paths and video bitrate which can provide the best QoE for users.

To better harness the benefit of multipath networks and increase network utilization, the agent also checks whether there exist paths being left idle when scheduling segment i. If so, the agent will schedule future video segments $(i+q)$ with the same bitrate level to be downloaded from these paths. This is to ensure that all paths are utilized to their full potential. In this paper, we set $q = 5$ empirically to download future video segments on spare paths.

IV. Evaluations

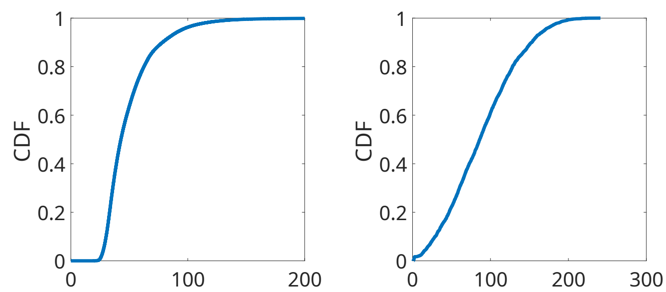
In this section, we evaluate the performance of the proposed joint decision-making algorithm in a controlled network environment with Mininet [14], followed by real-world experiments on a multipath testbed equipped with both broadband and satellite Internet. Both evaluations feature a similar network topology and application scenario. The implementation of this paper is available on GitHub[1].

A. Methodology

Application Scenario. In remote and rural areas, high-speed broadband Internet is usually not available and most existing satellite network service providers cannot guarantee high-speed and low-latency Internet services. Certain applications which require ultra-low latency Internet connections, such as online gaming and video conferencing suffer from poor QoE. This problem mainly results from the constrained bandwidth on broadband Internet and elevated latency on satellite Internet in these areas. In this work, we focused on VoD streaming services, which, unlike live video streaming, do not necessitate ultra-low latency connections. However, users' QoE can still be enhanced by leveraging the advantages offered by multipath scheduling algorithms.

We set up a Starlink satellite dish and monitored the network performance for consecutive three months. The Starlink dish has built-in network diagnostic metrics, which are accessible to users via gRPC interfaces. Fig. 1 shows the CDF of ping latency and downlink speed measured in ms and Mbps respectively. The ping latency is measured as the round-trip time (RTT) between the user dish and a Starlink Point-of-Presence (PoP) server connected to the nearest ground station.

[1]https://github.com/clarkzjw/globecom2023



(a) Ping latency (ms) (b) Downlink speed (Mbps)

Fig. 1: Starlink PoP Ping Latency and Speedtest CDF

As illustrated in Fig. 1, it can readily sustain an RTT under 100 ms and a high downlink throughput ranging from 100 to 150 Mbps.

Testbed Setup. We used *Picoquic* [10], which is a general QUIC implementation in C, as the transport layer library in our evaluation. *MABWiser* [21] is chosen as the contextual multi-armed bandit library to solve (7). It provides fast prototyping with various contextual multi-armed bandit algorithms. While other contextual multi-armed bandit algorithms can be used, in this work, we specifically evaluated the performance of two popular schemes, Linear Thompson Sampling (LinTS) [4] and Linear UCB (LinUCB) [16]. We implemented a custom video streaming client with the proposed joint decision-making algorithms in Python, which interacts with the *Picoquic* library through custom Python bindings.

To the best of our knowledge, our work is the first to address the multipath adaptive video streaming problem using a joint decision-making framework. Therefore, to benchmark and compare the video streaming performance with existing literature, we adopt certain simplifications. For previous independent decision-making algorithms, minRTT and Round-Robin (RR) are employed as the multipath scheduling algorithms. The bitrate adaptation in independent decision-making algorithms is achieved by selecting the next video segment with a bitrate closest to the estimated bandwidth $\mathrm{BW}_p(t)$ on each path p. We acknowledge that superior bitrate adaptation algorithms can be utilized. However, in this paper, we employ this simple bitrate adaptation scheme as a demonstration.

Videos. In this work, we recreated the *BigBuckBunny* video streaming dataset using a new bitrate ladder with high bitrate levels to meet the realistic network bandwidth availability as shown in Fig. 1. The *BigBuckBunny* video is segmented into 2-second segments and encoded with the bitrate ladder shown in Table II.

Metrics. For evaluating the performance of scheduling algorithms, the following metrics are used in our benchmark, average bitrate and rebuffering event counts. Both are essential metrics that greatly affect the QoE from users' perspectives. In our experiment, the video streaming playback lasts 400 seconds (200 video segments) in each round, and we repeated each experiment 20 times to obtain statistical results. To address the cold-start problem for CMAB algorithms, at the

TABLE II: Bitrate Ladder of The ABR Dataset

Resolution	Bitrate (Kbps)
1080p	87209.263
	71817.751
	55301.205
720p	35902.455
	22572.278
360p	4481.84

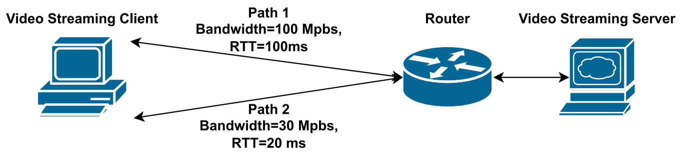


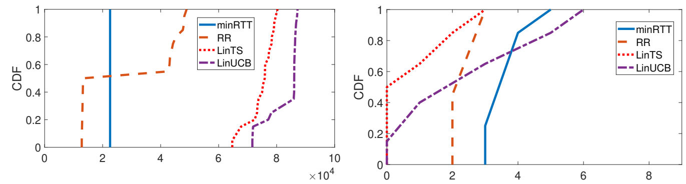
Fig. 2: Mininet emulation topology

beginning of each round, the agent explores each available arm once. The metrics collected during the initial exploration process are not included in Fig. 3 and Fig. 5.

B. Emulation Performance

To demonstrate the performance of the proposed joint decision-making algorithm in a controlled network environment, we utilized the Mininet emulation topology illustrated in Fig. 2. In alignment with the scenario discussed in Section IV-A, we mimic the satellite network connection with a high bandwidth and high latency link, while the broadband network link emulates a typical low latency and low bandwidth connection in rural areas. The bandwidth and latency parameters are set accordingly based on our Starlink measurement in Fig. 1 and generally available Internet services in rural areas [3]. The bandwidth and latency settings on both paths adhere to a normal distribution, with the mean values as depicted in Fig. 2. The router is connected to the streaming server via a high-speed data center connection, ensuring that there are no bottlenecks between the server and the router. Two independent network paths exist between a streaming client and the router. This topology represents a multi-homed scenario in which a client is connected through heterogeneous Internet services, specifically terrestrial broadband and satellite networks in this case. It also streamlines the commonly used CDN-based video streaming architecture, where each client requests video segments from different CDN servers based on client-side network measurement and adaptation. We assume that the video streaming client remains connected to the same CDN server throughout the entire video playback session. The potential competition for network resources among different LAN clients is not taken into account in this work.

Fig. 3 shows the QoE performance among four different scheduling algorithms. As shown in Fig. 3a, minRTT and RR cannot sustain high average playback bitrates compared with our joint decision-making algorithms. Among joint decision-making algorithms, LinUCB can achieve higher average playback bitrates compared with LinTS. However, it suffers more rebuffering events compared with LinTS as shown in Fig. 3b. LinTS also suffers fewer rebuffering events compared



(a) Average bitrate (Kbps)

(b) Rebuffering events

Fig. 3: QoE Performance with Mininet Emulation

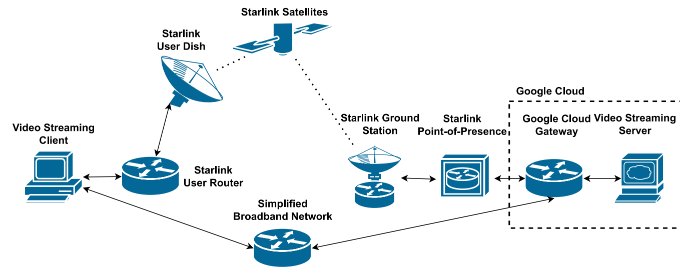


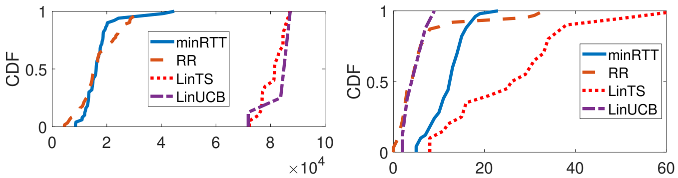
Fig. 4: Real-world experiment topology

with minRTT and RR. From the benchmark results obtained through Mininet emulation, it can be concluded that the joint decision-making algorithms offer improved QoE for streaming clients, in terms of both higher average playback bitrate and fewer rebuffering events.

C. Real World Experiment

To further support our conclusion from Mininet emulation, we built a multipath testbed on the Internet with a similar topology to conduct the same experiments. The topology is shown in Fig. 4. The nearest ground station to our on-site Starlink installation is located in Seattle, US. To minimize additional terrestrial networking latency, we deployed the video streaming server within Google Cloud's *us-west-1a* availability zone, physically located in Oregon, US. Based on our measurement, there is only one hop between the Starlink PoP server and Google Cloud's gateway server in this availability zone. Consequently, the impact of additional terrestrial networking latency can be minimized and we can assume there are no network bottlenecks between the Starlink PoP server and our video streaming server. The video streaming client is deployed on a Linux laptop, which is directly connected to the Starlink user router via Ethernet and to the terrestrial broadband Internet via Wi-Fi. To simulate the subpar Internet service quality in remote and rural areas, the network interface connected to the broadband Wi-Fi is managed using the Linux network utility, *tc*. We configure the latency and bandwidth on this interface both to follow a normal distribution with a mean value of 20 ms and 30 Mbps respectively.

Fig. 5 demonstrates the real-world performance benchmark of different scheduling algorithms. As illustrated in Fig. 5a, both LinTS and LinUCB outperform minRTT and RR in average playback bitrate, consistent with the Mininet emulation results. Moreover, in this case, LinUCB not only maintains the highest average playback bitrates but also experiences the



(a) Average bitrate (Kbps) (b) Rebuffering events

Fig. 5: QoE Performance with Real-World Experiment

fewest rebuffering events among all four algorithms, as shown in Fig. 5b.

V. Conclusions

In this paper, we introduce a novel joint decision-making scheduling framework for multipath adaptive video streaming, specifically in the context of VoD services. We merge the two independent control loops for multipath scheduling and video bitrate adaptation into a single joint decision-making process. The problem is modeled using contextual multi-armed bandit algorithms and addressed with a QoE-driven reward function design. The performance of the joint decision-making framework is evaluated using both Mininet emulation and a real-world multipath testbed. The application scenario considered in this paper combines terrestrial and satellite networks, which guarantee the independence of multi-network paths and assume there are no shared network bottlenecks between different paths. Other real-world scenarios where our approach is applicable include combined terrestrial and cellular networks. The results demonstrate that the proposed algorithm can deliver better QoE, including higher average bitrate and reduced rebuffering events when compared to independent decision-making algorithms. However, in this study, we only applied the scheduling algorithm to the granularity of video segments. The potential benefits of utilizing chunked CMAF encoding and transfer with contextual multi-armed bandit algorithms for ultra-low latency DASH can be further investigated.

Acknowledgment This work is supported in part by NSERC, CFI and BCKDF.

References

[1] Cisco annual Internet report (2018-2023) white paper. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] Dash reference player. [Online]. Available: https://github.com/Dash-Industry-Forum/dash.js

[3] NorthwesTel Internet plans in Whitehorse, Yukon. [Online]. Available: https://www.nwtel.ca/internet-plans

[4] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, no. 3, 2013, pp. 127–135.

[5] M. Bishop, "HTTP/3," RFC 9114, 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9114

[6] L. Cui, D. Su, S. Yang, Z. Wang, and Z. Ming, "TCLiVi: Transmission control in live video streaming based on deep reinforcement learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 651–663, 2021.

[7] R. Farahani, F. Tashtarian, A. Erfanian, C. Timmerer, M. Ghanbari, and H. Hellwagner, "ES-HAS: an edge- and SDN-assisted framework for HTTP adaptive video streaming," in *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2021, pp. 50–57.

[8] S. Ferlin, Ö. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," in *Proceedings of the 2016 IFIP Networking Conference and Workshops*, 2016, pp. 431–439.

[9] A. Ford, C. Raiciu, M. J. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 6824, 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc6824

[10] C. Huitema, "Picoquic." [Online]. Available: https://github.com/private-octopus/picoquic

[11] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9000

[12] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "MPTCP is not pareto-optimal: Performance issues and a possible solution," *IEEE Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.

[13] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1842–1866, 2017.

[14] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, pp. 1–6.

[15] L. Li, K. Xu, T. Li, K. Zheng, C. Peng, D. Wang, X. Wang, M. Shen, and R. Mijumbi, "A measurement study on multi-path tcp with multiple cellular carriers on high speed rails," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 161–175.

[16] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[17] Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, "ECF: An MPTCP path scheduler to manage heterogeneous paths," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, 2017, pp. 147–159.

[18] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," IETF, Internet-Draft draft-ietf-quic-multipath-05, 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/05/

[19] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE Transactions on Networking*, vol. 28, no. 4, pp. 1698–1711, 2020.

[20] T. Stockhammer, "Dynamic adaptive streaming over HTTP – standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*, 2011, pp. 133–144.

[21] E. Strong, B. Kleynhans, and S. Kadıoğlu, "MABWiser: A parallelizable contextual multi-armed bandit library for python," in *Proceedings of the 31st International Conference on Tools with Artificial Intelligence*, 2019, pp. 909–914.

[22] Y. Thomas, M. Karaliopoulos, G. Xylomenos, and G. C. Polyzos, "Low latency friendliness for multipath TCP," *IEEE Transactions on Networking*, vol. 28, no. 1, pp. 248–261, 2020.

[23] H. Wu, Ö. Alay, A. Brunstrom, S. Ferlin, and G. Caso, "Peekaboo: Learning-based multipath scheduling for dynamic heterogeneous environments," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2295–2310, 2020.

[24] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, and D. S. L. Wei, "An online learning assisted packet scheduler for MPTCP in mobile networks," *IEEE Transactions on Networking*, pp. 1–16, 2023.

[25] W. Yang, J. Cao, and F. Wu, "Adaptive video streaming with scalable video coding using multipath QUIC," in *Proceedings of the 2021 IEEE International Performance, Computing, and Communications Conference*, 2021, pp. 1–7.

[26] H. Yeo, C. J. Chong, Y. Jung, J. Ye, and D. Han, "NEMO: enabling neural-enhanced video streaming on commodity mobile devices," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[27] J. Zhao, J. Liu, C. Zhang, Y. Cui, Y. Jiang, and W. Gong, "MPTCP+: Enhancing adaptive http video streaming over multipath," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, 2020, pp. 1–6.