QoS-driven Contextual MAB for MPQUIC Supporting Video Streaming in Mobile Networks

Wenjun Yang, Lin Cai, *Fellow, IEEE,* Shengjie Shu,
Amir Sepahi, Zhiming Huang, and Jianping Pan, *Fellow, IEEE,*

Abstract—Video streaming performance may degrade substantially in a mobile environment due to fast-changing wireless links. On the other hand, to provide ubiquitous services, heterogeneous static and mobile access and backbone networks will be integrated in the sixth-generation (6G) systems, so mobile users can take advantage of multiple access options for better services. Multi-path transport-layer protocols like Multi-Path QUIC (MPQUIC) show promise in utilizing multiple access links to address the impact of mobility. However, the optimal link selection that aims to provide statistical QoS guarantee for video streaming in a mobile environment with both user mobility and network mobility remains an open issue. In this paper, based on a lightweight Multi-Armed Bandit (MAB) technique, we develop a QoS-driven Contextual MAB (QC-MAB) framework for MPQUIC, which makes an intelligent access network selection and adaptively enables FEC coding to trade off delay, reliability and goodput. Extensive simulation results with ns-3 show that the proposed QC-MAB framework can outperform the state-of-the-art solutions. It achieves up to ten times lower video interruption ratio and three times higher goodput in highly dynamic mobile environments.

Index Terms—Multipath QUIC, mobility, video streaming, MAB, network selection, FEC

1 Introduction

We have witnessed a proliferation of video streaming applications on mobile devices. The use of mobile phones for video conferencing is convenient for people outside of their offices. By accessing real-time video from roadside units, self-driving vehicles can be aware of what is happening behind corners and avoid dangerous blind spots.

To handle network dynamics while maintaining a reasonable user experience, scalable video coding (SVC) has been proposed and adopted to encode video once and decode the bitstream multiple times with different resolutions, frame rates, and video qualities. However, using single-path transmission is quite hard to provide high-quality video services for high-mobility users due to the dynamic channel quality of the access link [1]–[3].

Consequently, multipath video streaming using multiple access links has been extensively investigated in recent years. MPDASH [4] introduces a deadline-aware multipath scheduler into Dynamic adaptive streaming over HTTP (DASH) [5] and takes user preferences into account. However, network states are time-varying especially when mobility is introduced, so MPDASH suffers from performance degradation in dynamic systems. Since deep Reinforcement Learning (DRL) is a powerful tool to foresee the variations of network states [6]–[9], there are many DRL-based solutions, such as Meta-DAMS [10], DRL360 [11] and Peekaboo [12].

In addition to learning-based multipath scheduling, when Forward Error Correction (FEC) is used in conjunction with multipath transmission, the on-time packet delivery reliability can be substantially enhanced [1], [13]–[15]. ADMIT [1] integrates FEC coding and rate allocation over multiple paths to maximize mobile video quality using multipath TCP (MPTCP). In order to achieve QoS guarantee rather than merely improving QoS, LEAP [15] splits data into multiple subflows according to the data flow's QoS requirements, as well as utilizing cross-path FEC coding to trade off between throughput, delay deadline, and reliability.

However, in mobile environments, the existing learning-based or FEC-combined multipath solutions encounter new challenges. Peekaboo faces challenges in effectively dealing with the highly dynamic end-to-end (E2E) path state (context) in mobile environments, which means the dimension of context keeps increasing as time passes. Therefore, the dependencies between context and QoS performance may take considerable time to discover. Both Peekaboo and LEAP fail to fully explore the diversity of access networks and do not proactively switch to a better one during user movements, leading to significant performance degradation.

Based on the principle that the end system can take the input from the network to optimize the configuration of the transport protocol [16], in this paper, we develop a QoS-driven Contextual MAB (QC-MAB) framework for MPQUIC to support video streaming in mobile networks. The main contributions are three-fold.

- QC-MAB provides insights into offering a statistical QoS guarantee under the impact of mobility. Unlike existing solutions that focus on achieving a deterministic delay guarantee, which is challenging, we explore how to control the percentage of overdue frames within a tolerable threshold while also striving for high video quality.
- To improve learning efficiency, we design a context refinement strategy consisting of two steps: context

- *W. Yang, L. Cai, and A. Sepahi are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8P 5C2, Canada. S. Shu, Z. Huang, and J. Pan are with Department of Computer Science, University of Victoria, Victoria, BC V8P 5C2, Canada.*
  E-mail: {wenjunyang, shengjies, amirsepahi, zhiminghuang, pan}@uvic.ca, cai@ece.uvic.ca

dimension reduction and context noise removal. By finding correlations among different features observed by the agent, we integrate multiple correlated features into one to reduce the context dimension, thereby reducing the learning time for determining the reward given the context. In addition, we utilize the Adaptive Kalman Filter (AKF) and exponentially weighted moving average (EWMA) approaches to remove the noise of the observed features so that the best arms can be found quickly.

- We implemented QC-MAB and the benchmark algorithms for MPQUIC in network simulator 3 (ns-3). Based on the implementation, we run the video data extracted from a real movie to validate the performance of our proposal in terms of video interruption ratio (IR), mean video quality, E2E delay for all BL packets, and the aggregated goodput. The experimental results demonstrate that QC-MAB outperforms the benchmark algorithms to achieve up to ten times lower IR and three times higher goodput in highly dynamic mobile environments.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 presents the system model and QoS guarantee problem. Section 4 elaborates on the detailed design of QC-MAB. Section 5 evaluates the proposed algorithm via simulations, followed by concluding remarks and further research issues in Section 6.

2 Related Works

The development of video bitrate adaptation and multipath transmission has greatly pushed forward video streaming.

Video rate adaptation. DASH [5] is agnostic of the video codec and can use any codec including H.264/SVC [17] and H.265 [18]. It provides the opportunity to adjust the video bitrate during playback for high QoE. With SVC, we can encode video once and decode the bitstream multiple times with different resolutions, frame rates, and video quality [19], [20], so the encoding time and server storage space can be saved, which is particularly important for live or high-quality stored video streaming. There are also approaches for optimizing fixed QoE objectives based on playback statistics such as buffer occupancy [21], or for altering QoE goals based on control theory [22] or reinforcement learning [7], [11]. Recently, [23] designed a decision-tree-based throughput predictor, named Lumos, to select the bitrate of video chunks based on the network capacity. However, these approaches focus on scenarios with single-path transmission. They are sensitive to network uncertainties.

Multipath transmission. Multipath transport protocols, such as multipath TCP (MPTCP) [24] and MPQUIC [25] have many potentials, such as bandwidth aggregation, high-reliability provisioning, seamless handover, etc. Various multipath schedulers [24], [26]–[29] based on MPTCP or MPQUIC are proposed for different scenarios and objectives. However, they do not consider the video QoS requirements in their design.

The recent efforts for QoS provisioning with multipath delivery can be divided into: 1) estimation model-based, e.g., [4], [30]–[32], 2) learning-based policies, e.g., [6], [8], [9], [12], [33], and 3) FEC-combined multipath schemes, e.g., [1], [13]–[15].

CMT-DA [30] utilizes the path status estimation to adjust sending rate across multiple paths to minimize the E2E video distortion. MPDASH [4] takes user preferences into account, scheduling packets over multiple paths following the throughput estimation to meet the user’s deadline or priority request. Both DEMS [31] and DAMS [32] estimate RTT to schedule the sending order of video blocks with the consideration of the block’s deadline.

However, the above solutions are sensitive to sudden network condition changes which may occur frequently in mobile environments, so the learning-based approaches have been investigated. [6] formulated the video streaming process over multiple links as a reinforcement learning (RL) task to meet QoS goals in heterogeneous networks. ReLeS [8] formulated the multipath scheduling problem as an RL task with the consideration of QoS characteristics under various network scenarios. PERM [9] employed an actor-critic network to optimize the multi-path packet scheduling and bitrate adaptation simultaneously. Nevertheless, RL/DRL-based techniques increase the algorithm’s reaction time dramatically [34]. Therefore, the lightweight contextual multi-armed bandit (CMAB) [12] framework is more desirable for applications with stringent QoS concerns. Alzadjali et al. [33] proposed an online MPTCP path manager based on the CMAB to help choose the optimal primary path connection that maximizes throughput and minimizes delay and packet loss. Peekaboo [12] is a novel CMAB-based MPQUIC scheduler to cope with dynamically changed channel conditions of the heterogeneous paths.

On the other hand, in the context of multipath, FEC coding can offer a great chance to balance the tradeoff among QoS metrics. For instance, a Reed-Solomon (RS) code is employed in ADMIT [1], a rateless Raptor code in FMTCP [13], and a systematic random linear code in SC-MPTCP [14]. These works demonstrated that FEC is well suited to maximizing goodput without violating delay requirements. The closest work to our focus on QoS guarantee is LEAP [15] which uses RS code in conjunction with adaptive congestion control over multiple paths to meet the QoS requirements in terms of throughput, delay, and reliability.

In a nutshell, all solutions above have direct or indirect positive impacts on the QoS performance of video streaming with multipath delivery. Yet none of them takes into account the challenges and opportunities in mobile environments.

Mobility in 6G is ubiquitous and diverse. Not only end users but also backbones (e.g., satellite networks) and access networks (e.g., UAV) could be mobile, which leads to the fast-changing traffic density of each access network and the highly dynamic congestion level of core networks. Both the estimation model-based and learning-based solutions lose accuracy if they fail to consider the dynamics caused by mobility.

The access networks visible to an end host are changing all the time along with movements, which implicitly offers a good opportunity to meet stringent QoS requirements if the end host proactively switches to the access networks with better characteristics.

Even though MMSPEED [35] and [36] focused on QoS guarantee with multipath delivery in mobile environments,

This article has been accepted for publication in IEEE Transactions on Mobile Computing. 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/TMC.2024.3507051

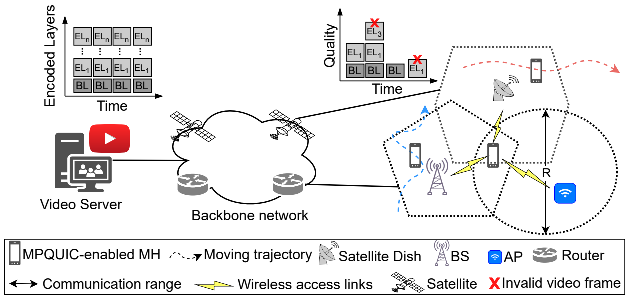


Figure 1: Overview of multipath video transmission in mobile networks.

they differ fundamentally from our work because they targeted sensing data exchange in wireless sensor networks, using the link-layer approach rather than the transport-layer one.

Motivated by mobility challenges, in this paper, we investigate the impact of mobility on QoS performance for video streaming over MPQUIC.

3 System Model

In this section, we introduce the background information and define the system model for the problem investigated. We first introduce the mobile access network model, followed by modeling the packet transmission over multiple E2E paths. Finally, we discuss the crucial QoS metrics for video streaming applications.

3.1 Wireless Access Network Model

We consider a scenario where $\mathcal{M}$ access networks are densely distributed over a certain area, as shown in Fig. 1. Different types of access points, e.g., LTE/5G base station (BS), WiFi access point (AP), and UAV/satellite AP, have different communication channels associated with different data rates, communication ranges, etc. For simplicity, we use AP to represent all kinds of access points hereafter.

An MPQUIC-enabled mobile host (MH) is requesting video service from the video service provider through multiple access networks simultaneously. MH moves in the Random Walk Mobility Model (RWMM) manner, i.e., moves along a direction $\theta \in [0, 2\pi)$ with a constant velocity within a short time slot. At the end of each time slot, the direction is reshuffled, independent from slot to slot and node to node. If MH moves out of the communication range of the currently serving AP, known as a service outage, a handoff occurs so that the MH can be served by another AP.

3.2 Multipath Transmission Model

A library $\mathcal{F} = \{1, \ldots, f, \ldots, F\}$ of video frames is stored in the video server. With certain coding techniques, such as SVC, each frame f is encoded in several layers to provide different qualities. The example in Fig. 1 shows the layered coding: base layer (BL) and enhancement layers (ELs). The BL contains the low-resolution information of each frame and renders the video with the minimum quality, and each EL is to provide higher video quality on the top of the lower layers.

The encoded F frames contain a set of packets with distinct significance. Let $\mathcal{K} = \{1, \ldots, k, \ldots, K\}$ denote the packet sequence number for all F frames. Denote by L_k the layer type associated with packet $k \in \mathcal{K}$. Specifically, $L_k = 0$ if k carries the BL data while $L_k = 1$ if k carries the EL data. Given that the MH is equipped with P interfaces, by which P MPQUIC connections (a.k.a. subflows or E2E paths) can be established between the video server and the MH. Let $\mathcal{P} = \{1, \ldots, p, \ldots, P\}$ be a set of available paths. Each path p independently maintains several time-varying conditions: $w_p(t)$, $RTT_p(t)$ and $\epsilon_p(t)$, which refer to the congestion window (cwnd), round-trip-time (RTT) and packet loss rate at time t, respectively. For simplicity, we ignore the notation t hereafter. With a certain scheduling policy, a portion of packets will be allocated into each path until all path windows are fully filled. A new round of data allocation will begin over the paths receiving a new acknowledgment. This scheduling process repeats until all K packets are sent.

Using FEC, a block containing n packets is encoded into a longer block with $\hat{n}$ packet, where $n/\hat{n}$ is the coding rate. It is guaranteed that any n of the $\hat{n}$ encoded packets can recover the original data in the n packets. Therefore, if the number of packet losses is no larger than $\hat{n} - n$, the receiver can successfully decode the data without retransmission. Given an MPQUIC path p with a packet loss probability ϵ_p, and let l_p be the decoding failure probability for the block b from path p, we have [13]

$$l_p = \sum_{j=\hat{n}-n+1}^{\hat{n}} \left(\begin{array}{c} \hat{n} \\ j \end{array} \right) \epsilon_p^j \left(1 - \epsilon_p\right)^{\hat{n}-j}. \tag{1}$$

3.3 Video Quality Model

The minimum acceptable video quality is free of playback interruption [6]. Beyond that, the higher the video resolution, the better the user experience.

To meet minimum video quality requirements, it is crucial for video clients to receive BL packets on time, ensuring that each packet arrives within a specified time constraint referred to as ΔT. However, in highly dynamic systems, offering a deterministic delay guarantee can be a challenging task. Hence, we delve into the concept of a statistical delay guarantee, which is quantified by the reliability Γ. This reliability signifies the probability of successfully achieving the on-time guarantee for BL packets. Furthermore, it is desirable to achieve higher goodput Φ in order to facilitate the delivery of a larger number of EL packets.

Therefore, the video quality depends on E2E delay, reliability, and goodput.

3.3.1 E2E Delay

Assuming that packet k is scheduled into a path $p^* \in \mathcal{P}$, the application-level E2E latency of packet k denoted as d_k^E can be divided into two main components: network delay (d_k^N) and transport-layer delay (d_k^T). In other words, we have $d_k^E = d_k^N + d_k^T$.

d_k^N includes all of the transmission delay, propagation delay, processing delay and queuing delay along the path when the packet traverses from the source to the destination, referred to as one-way delay (OWD) τ_{p^*}, which is approximately half of the RTT_{p^*}. Note that the transmission delay of the wireless uplinks or downlinks here is uncertain due to user and/or network mobility. As video data from the server to the mobile user go through downlinks, we focus on downlink performance. Let C_{p^*} be the transmission rate of wireless downlinks, we have

$$C_{p^*} = \xi W_m \log_2(1 + SINR_m), \tag{2}$$

where $SINR_m$ stands for the received signal to interference and noise ratio of the m-th AP, W_m is the corresponding channel bandwidth, and $\xi \in (0, 1)$ is an efficiency coefficient of the communication system, depending on several factors, e.g., hardware and software design. Therefore, the transmission delay of packet k over the wireless downlinks between the m-th AP and the MH is x_k/C_{p^*} where x_k is the size of packet k.

d_k^T consists of retransmission delay ζ_k and reordering delay δ_k. ζ_k is caused by packet loss events, depending on the congestion control phase, as discussed in [24]. δ_k is due to the out-of-order (OFO) situation, so it has an upper bound based on the maximum difference between the one-way packet delay on routes, that is, $0 \le \delta_k \le \max_{p\in\mathcal{P}} \tau_p - \tau_{p^*}$.

3.3.2 Reliability

Considering the feature of SVC video traffic, reliability is defined as the probability of on-time delivering of BL packets, denoted by γ_k. γ_k can be expressed as $\gamma_k = \mathbb{P}\{d_k^E \le \Delta T\}$.

Without FEC applied, the reliability of BL packets largely depends on the path error rate including congestion loss and wireless link errors. As long as the time constraint ΔT has not expired, each lost packet can be retransmitted in the subsequent round trips to improve its successful delivery probability. Assuming that packet losses are independent, the process can be represented by a Bernoulli trial. Let $V \in \mathbb{Z}_0^+ = \{0, 1, \cdots\}$ be the maximum retransmission times that can take place before ΔT expires, we have

$$\gamma_k = \sum_{v=0}^{V} (\epsilon_{p^*})^v (1 - \epsilon_{p^*}).$$

Note $v = 0$ refers to the successful packet delivery in the initial transmission.

Similarly, when FEC is applied, γ_k depends on the decoding failure probability l_{p^*} as described in (1), and it can be estimated as:

$$\gamma_k = \sum_{v=0}^{V} (l_{p^*})^v (1 - l_{p^*}).$$

3.3.3 Goodput

By definition, the aggregated goodput denoted by Φ stands for the number of useful information bits delivered by multiple paths to the destination per unit of time. Let $\mathcal{J}$ be a subset of $\mathcal{K}$, storing all received decoded packets from multiple paths within the time interval Δt, Φ is expressed as

$$\Phi = \frac{|\mathcal{J}| \cdot MSS \cdot 8}{\Delta t}. \tag{3}$$

Here $|\cdot|$ stands for the number of elements inside a set and MSS means the maximum segment size in bytes.

In the context of multipath transmission, the goodput perceived by the user is the summation of the goodput of each path. Let ν_p be an indicator of whether the FEC is enabled or not. ν_p equals 1 if FEC is enabled and 0 otherwise. Denote by $\phi_p(\nu_p)$ the goodput on path p, it can be calculated as the product of the transmission rate and the fraction of packets successfully received when $\nu_p = 0$, while it is calculated by subtracting the overhead due to the transmission of redundant packets from the total transmission rate when $\nu_p = 1$.

Let $\phi_p(\nu_p)$ represent the goodput on path p. When $\nu_p = 0$, it is calculated as the product of the transmission rate and the fraction of packets successfully received. When $\nu_p = 1$, the goodput is calculated by subtracting the overhead from transmitting redundant packets from the total transmission rate. Thus, $\phi_p(\nu_p)$ is given by

$$\phi_p(\nu_p) = \begin{cases} \frac{w_p\cdot(1-\epsilon_p)}{RTT_p}, & \text{if } \nu_p = 0 \\ \frac{w_p\cdot(1-l_p)\cdot(n/\hat{n})}{RTT_p}, & \text{if } \nu_p = 1 \end{cases} \tag{4}$$

Φ is therefore expressed as

$$\Phi = \sum_{p\in\mathcal{P}} \phi_p. \tag{5}$$

4 QC-MAB for Multipath Video Streaming

In this section, we first formulate the problem we are focusing on, followed by the details of our solutions.

4.1 Problem Formulation

Given the QoS requirements $\langle \Delta T, \Gamma \rangle$ as described above, our main objective is to determine a policy π that effectively selects an access network for each E2E path and dynamically enables or disables the FEC technique to fulfill the delay and reliability criteria, while simultaneously maximizing the goodput. This can be formulated mathematically as follows:

$$\begin{aligned} \max_{\pi} \quad & \Phi \\ \text{s.t.} \quad & \nu_p \in \{0, 1\} \\ & \phi_p(\nu_p) \le C_p, \quad \forall p \in \mathcal{P} \\ & \gamma_{k_b} \ge \Gamma, \quad k_b \in \{j | L_j = 0, j \in \mathcal{J} \subseteq \mathcal{K}\}. \end{aligned} \tag{6}$$

Because both integer variables and continuous variables are used, the problem in (6) is a mixed-integer programming (MIP) problem, which is challenging to solve with mathematical optimization methods. It has been proven that reinforcement learning (RL) is a promising tool to solve MIP problems [37]. Therefore, one intuitive idea is to leverage RL-based approaches to discover the relationship between network environments and QoS performances from the historical data samples, and thus make an intelligent decision to meet the user's QoS requirements. MAB is a simplified version of RL. Compared to such RL algorithms as Q-learning and actor-critic, MAB neither involves the state transition nor considers a sequence of actions and their effects on the environment, and the agent makes decisions solely based on the actions taken and the rewards received. As a result, the decision can be made at a milliseconds level. For video streaming applications that demand stringent delay guarantees in highly mobile systems, the simple yet efficient MAB is preferable.

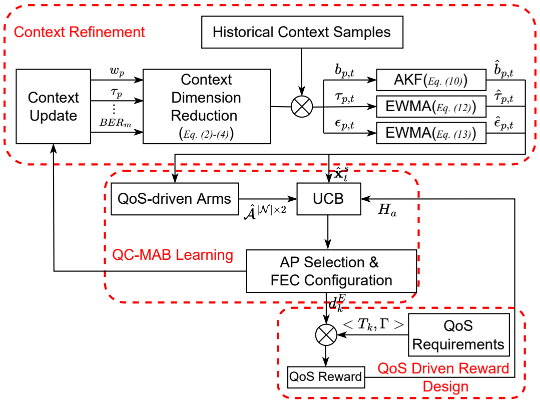


Figure 2: QC-MAB framework.

Table 1: Context space design

E2E path related context
w_p: the cwnd on path p
Ψ_p: bytes-in-flight on path p
τ_p: OWD of path p
ϵ_p: packet loss rate on path p

Wireless links related context
$(\hat{x}, \hat{y})$: MH's location
(x_m, y_m): the m-th AP's location
P_m: received power
σ: background noise
I_m: the interference
W_m: the available bandwidth
BER_m: the link bit error rate

In this section, we propose a QC-MAB framework to solve the QoS guarantee problem, with which the underlying relationship between context and QoS requirements can be discovered without a lengthy training process, thereby facilitating decision-making in highly dynamic systems.

Fig. 2 depicts our proposed QC-MAB framework for MPQUIC-based video streaming. First, three components of QC-MAB are designed: Context refinement, QoS-driven arms design, and QoS-driven rewards design. Then, a contextual bandit problem is formulated, and the corresponding solutions are discussed.

4.2 Context Refinement (CR)

The networking features that affect the QoS requirements are collectively referred to as context, known as state s. We use context and state interchangeably throughout this paper. Each state s is defined by a vector of features $\mathbf{x}^s = (x^s_{E_1}, \cdots, x^s_{E_c}, x^s_{L_1}, \cdots, x^s_{L_d})$ where $(x^s_{E_1}, \cdots, x^s_{E_c})$ represents the E2E path relevant features, and $(x^s_{L_1}, \cdots, x^s_{L_d})$ means features related to the last-mile wireless links. According to the analysis in the last section, 11 critical features related to QoS on path p are summarized in Table 1. In this case, $\mathbf{x}^s$ is an 11×1 vector.

However, with a large dimension of contexts, discovering the relationship between context and rewards takes a long time. To mitigate this issue, we try to reduce the dimension of $\mathbf{x}^s$ by finding correlations among features and then integrating several contextual features into one. Furthermore, most contexts are estimated or measured values based on instantaneous feedback from the network, which can be noisy due to sudden changes in network conditions. In order to reveal the intrinsic relationship between contexts and rewards, we need to filter out the noise within contexts. Context dimension reduction and noise removal are referred to as context refinement (CR).

4.2.1 *Context Dimension Reduction*

w_p and Ψ_p jointly reveal the congestion situation and bottleneck capacity C_p^N at the network side, denoted by $C_p^N := f(w_p, \Psi_p)$. On the other hand, $(\hat{x}, \hat{y})$, (x_m, y_m), P_m, σ_m, I_m and W_m define the achievable access link capacity C_p^A of path p as follows according to Eq. (2),

$$C_p^A = \xi W_m \log_2\left(1 + \frac{P_m\left(\sqrt{[\hat{x} - x_m]^2 + [\hat{y} - y_m]^2}\right)^{-e}}{\sigma_m^2 + I_m}\right), \quad (7)$$

where e is the path loss exponent.

Therefore, we define a new context b_p, representing the bottleneck bandwidth of path p. This context is derived by integrating C_p^N and C_p^A in the following manner:

$$b_p = \min\{C_p^N, C_p^A\}. \quad (8)$$

In addition, the relationship between the packet loss rate ϵ_p and the link bit error rate BER_m can be expressed with [1]

$$\epsilon_p = \rho + (1-\rho)\left(\sum_{i=a}^{Y} \binom{Y}{i} (BER_m)^i (1 - BER_m)^{L-i}\right), \quad (9)$$

where ρ stands for the loss rate caused by network congestion, Y is the length of each packet in bits, and a is the minimum number of bits in error that will lead to packet error due to limited error coding capacity in the physical and link layers.

Under the assumption that every ΔT the MH displays a new frame and runs QC-MAB[1], the time at the receiver side can be slotted by $\mathcal{T} = \{1, 2, \ldots, t, \ldots\}$ where t means the t-th display event. Since the correlation among each context can be realized using (7)–(9), the context space observed by the MH is reduced to a set of three parameters at slot t, i.e., $\mathbf{x}_t^s = (b_{p,t}, \tau_{p,t}, \epsilon_{p,t})$. $b_{p,t}$ and $\tau_{p,t}$ can be directly measured by the learning agent at the MH. In mobile environments, the mobile access link BER far outweighs congestion losses within backbone networks [38]. Hence, $\epsilon_{p,t}$ is roughly equal to BER which can be also directly estimated at the MH side. Meanwhile, the sender will periodically notify the receiver of the estimated loss event rate by measuring the packet loss event rate. Therefore, in the case where backbone networks might be the bottleneck and the congestion loss dominates $\epsilon_{p,t}$, the receiver can use the sender's loss event calculation to estimate $\epsilon_{p,t}$.

4.2.2 *Context Noise Removal*

In reality, a common approach to obtain $b_{p,t}$ feedback is to monitor the timestamps of the received consecutive packets to estimate bottleneck bandwidth. Assuming S_t and S_{t-1} are the total amounts of data received by the MH at successive measurement time epochs t and $t-1$, $b_{p,t}$ is measured with

$$b_{p,t} = \frac{S_t - S_{t-1}}{\Delta t}. \quad (10)$$

[1] Compared to the video server, the MH has easier access to collect the necessary features locally. This is why we choose to run the QC-MAB at the MH side. In our design, the QC-MAB handles access network selection for the MH and FEC configuration for the video server. By running QC-MAB at the MH, it becomes feasible for the MH to send an acknowledgment (ACK) along with the decision regarding FEC configuration during the subsequent round of transmission back to the video server.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. 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/TMC.2024.3507051

However, the measured $b_{p,t}$ in this manner is likely biased due to many reasons, e.g., the sending window is too small during slow startup to match the actual network capacity, so the instantaneous $b_{p,t}$ is much lower than it should be. Therefore, we utilize an Adaptive Kalman Filter (AKF) [39] to remove the noise within $b_{p,t}$. Given limited measured samples, AKF can still quickly narrow down to the truth by taking a few of those inputs and beyond by understanding the variation or the uncertainty of those inputs. The standard Kalman filter goes through two stages to acquire the final estimate as follows.

The first stage is *prediction*, in which the estimated capacity (b_{p,t_e}) of path p and the estimation errors $Z_{p,t}$ are updated as

$$\begin{aligned} b^-_{p,t_e} &= b^+_{p,t-1} + \omega_{p,t}, && (11)\\ Z^-_{p,t} &= Z^+_{p,t-1} + Q_p, && (12) \end{aligned}$$

where the superscript "+" indicates that the estimate is a posterior, and "-" indicates a prior estimate. $\omega_{p,t}$ stands for white Gaussian noise with zero mean and variance Q_p, which depends on network uncertainties such as user mobility.

The second stage is *correction*. The measured network capacity is assumed disturbed by a white Gaussian noise $\upsilon_{p,t}$ with zero mean and variance R_p, such that

$$y_{p,t_m} = b_{p,t_m} + \upsilon_{p,t}. \tag{13}$$

Here b_{p,t_m} is the instantaneously measured value at time t, using (10). Then the critical Kalman gain is derived by

$$K_{p,t} = Z^-_{p,t}\left(Z^-_{p,t} + R_p\right)^{-1}, \tag{14}$$

with which we can determine whether the predicted value or measured value is closer to the true value. Let $\hat{b}_{p,t}$ and $\hat{Z}_{p,t}$ be the final correction for the predictions of b^-_{p,t_e} and $Z^-_{p,t}$, we have

$$\begin{aligned} \hat{b}_{p,t} &= b^-_{p,t_e} + K_{p,t}\left(y_{p,t_m} - b^-_{p,t_e}\right), && (15)\\ \hat{Z}_{p,t} &= Z^-_{p,t} - K_{p,t} Z^-_{p,t}. && (16) \end{aligned}$$

The corrected $\hat{b}_{p,t}$ and $\hat{Z}_{p,t}$ will be the posterior states that are involved in the future rounds of corrections. In this way, the estimation error diminishes over time, and so does the noise in $b_{p,t}$[2].

Finally, we employ the exponentially weighted moving average (EWMA) approach to correct measured $\tau_{p,t}$ and $\epsilon_{p,t}$ as follows:

$$\begin{aligned} \hat{\tau}_{p,t} &= E_1(t) = \alpha \cdot \tau_{p,t} + (1-\alpha)\cdot E_1(t-1), && (17)\\ \hat{\epsilon}_{p,t} &= E_2(t) = \beta \cdot \epsilon_{p,t} + (1-\beta)\cdot E_2(t-1). && (18) \end{aligned}$$

The settings of α and β are discussed in Section 5.4.

Then, the final context $\hat{\mathbf{x}}^s_t = \{\hat{b}_{p,t}, \hat{\tau}_{p,t}, \hat{\epsilon}_{p,t}\}$ are obtained.

4.3 QoS-Driven Arms Design

To find the optimal policy π as presented in problem (6), we design arms, a.k.a., actions, from two aspects: access network selection and FEC configuration. The former is for the MH and the latter one is for the video server. As a result, each arm a is associated with a vector of behaviors $\mathbf{x}^a = (x^a_1, x^a_2)$ where x^a_1 is related to the network selection and x^a_2 is associated with FEC option. Given there are $\mathcal{M}$ access networks to be selected, and two options of whether enabling FEC, the set of arm vectors, $\mathcal{A}$, includes $|\mathcal{M}| \times 2$ combinations. Similarly, the larger number of arms, the longer it will take to explore the arms that are never or rarely chosen. Therefore, the arm set is refined by filtering out those arms that do not meet specific criteria. For example, for those paths whose $\hat{\tau}_{p,t}$ is greater than ΔT, we can skip the choice of its associated access network. Therefore, the new arms space $\hat{\mathcal{A}}^{|\mathcal{N}|\times 2}_t$ is yielding ($|\mathcal{N}| \leq |\mathcal{M}|$).

4.4 QoS-Driven Rewards Design

In the QC-MAB framework, the reward function is designed according to the QoS requirements. Fulfilling a deterministic QoS guarantee (e.g., always less than 100 ms delay across a network) is generally hard in highly dynamic systems, so we are motivated to achieve a statistical QoS guarantee (e.g., achieving less than 0.001% packet overdue) with a proper reward function design for video streaming applications.

Specifically, we assume each video frame f contains N of packets, which are categorized into n_b BL packets and n_e EL packets. Therefore, the on-time delivery ratio for BL packets within frame f can be calculated by $\gamma_f = \frac{\sum_{i=1}^{n_b} \mathbb{1}\{d^E_{b_i} \leq \Delta T\}}{n_b}, b_i \in \mathcal{K}$. Note $\mathbb{1}\{\cdot\}$ refers to indicator function. Provided that the video client's reliability preference is Γ for BL. If γ_f is greater than Γ, we give a weighted bonus to packets that meet their individual deadline. In this way, the learning agent will strive to guarantee the on-time delivery of BL packets while sending as many EL packets as possible.

Otherwise, if the on-time reliability requirement is not satisfied, the reward for EL packets will be discounted by a factor η, motivating the learning agent to choose some actions (e.g., enable FEC) to reduce the E2E delay for BL packets. Here $\eta = \frac{1}{n_b - \sum_{i=1}^{n_b} \mathbb{1}\{d^E_{b_i} \leq \Delta T\}}$, implying that the more expired BL packets there are, the less useful the EL packets become.

Let $r|(s,a)$ be the posterior reward r which is parameterized by state s and action a, we have (19). It is worth noting that the reward is designed for a whole frame instead of a single packet due to the strong correlation between packets within the same video frame, in other words, a reward will be recalculated upon each frame completion.

4.5 Contextual Bandit Problem

With the QC-MAB framework, the QoS guarantee issue can be formulated into a cumulative rewards maximization problem, known as the contextual bandit problem formulated in **P1**. **P1** indicates that our goal is to find a good

[2] To be able to use the Kalman filter, it is necessary to have the knowledge of the co-variances of the prediction and measurement noise, i.e., Q_p and R_p. However, they are unknown a priori in our case, and they may vary over time as the network environment changes. To address this issue, we can use Least Squares method to estimate them from the innovation signal's autocorrelation as in [40].

$$r|(s,a) = \begin{cases} \left(\sum_{i=1}^{n_b} \mathbb{1}\{d_{b_i}^E \leq \Delta T\} + \sum_{j=1}^{n_e} \mathbb{1}\{d_{e_j}^E \leq \Delta T\}\right)/N, & \text{if} \frac{\sum_{i=1}^{n_b} \mathbb{1}\{d_{b_i}^E \leq \Delta T\}}{n_b} \geq \Gamma \\ \left(\sum_{i=1}^{n_b} \mathbb{1}\{d_{b_i}^E \leq \Delta T\} + \eta \sum_{j=1}^{n_e} \mathbb{1}\{d_{e_j}^E \leq \Delta T\}\right)/N, & \text{otherwise,} \end{cases} \tag{19}$$

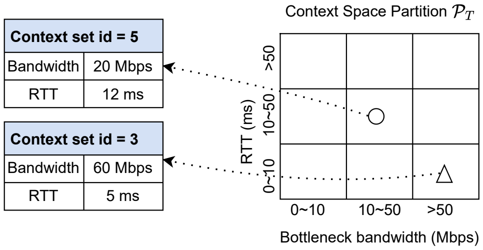


Figure 3: Illustration of context space partition and the context set ID identification.

policy π that determines the action a_t given the state s_t to maximize the expected cumulative rewards.

$$\textbf{P1:} \max_{\pi: s_t \to a_t} \mathbb{E}\left[\sum_{t=1}^{T} [r_t|(s_t, a_t)]\right]. \tag{20}$$

There are many existing algorithms to solve the MAB problems. The Upper Confidence Bound (UCB) [41] is a lightweight yet efficient bandit algorithm that is proven to produce asymptotically optimal regret performance. Therefore, we incorporate the UCB algorithm into QC-MAB to solve **P1**. Every ΔT the MH displays a new frame, it activates QC-MAB to calculate rewards and proactively select APs for connection. Afterward, the server is notified of the FEC enabling decision along with an acknowledgment, and then it configures FEC accordingly for the next round of data transmission.

4.6 UCB Solution

Denote by $\mathcal{S}$ the set of context vectors $\hat{\mathbf{x}}_t^s$ over time. Because context features in mobile environments are changing over time, the dimension of $\mathcal{S}$ may increase unbounded as time passes, which poses challenges to learning the relationship between contexts and rewards.

To maintain the context space in an acceptable finite dimension, we create a partitioning rule $\mathcal{P}_T$, which maps $\hat{\mathbf{x}}_t^s$ at each time step to an integer value. Let D be the dimension of $\hat{\mathbf{x}}_t^s$, and h_d be the number of partitioned slices in the feature domain where $d \in [1, D]$. Therefore, the dimension of the context space $\mathcal{S}$ can be at most $\prod_{d=1}^{D} h_d$. Fig. 3 gives an example of a 2-dimensional context partition, i.e., $\hat{\mathbf{x}}_t^s = \{\hat{b}_t, \hat{\tau}_t\}$. To map the pair of $\hat{b}_{p,t}$ and $\hat{\tau}_{p,t}$ to a certain value, we respectively apply two thresholds: 10 Mbps and 50 Mbps for the $\hat{b}_{p,t}$, and 10 ms and 50 ms for the $\hat{\tau}_{p,t}$, to slice the bandwidth and RTT domains. Therefore, bandwidth values fall within three slices: 0–10, 10–50, and greater than 50. Similarly, all RTT values fall within three segments. In this case, $h_1 = h_2 = 3$ and there are 3×3 combinations across these two domains, which means any pair of $\hat{b}_{p,t}$ and $\hat{\tau}_{p,t}$ can be mapped into an integer between 1 and 9.

For each 3-dimensional vector $\hat{\mathbf{x}}_t^s \in \mathcal{S}$, it can be mapped to a tuple $q_t \in \mathcal{P}_T$. At the very beginning ($t = 1$), the context is mapped to tuple q_1 and the reward distribution on each arm is unknown yet, so we assume the mean reward $\bar{r}_{a_1|q_1}$ for all arm $a_1 \in \hat{\mathcal{A}}_1$ is identical. The learning agent starts with an arm a_1^* randomly.

Upon a new frame display event at time t ($t \geq 2$), the learning agent first calculates the conditional reward $r_{a_{t-1}^*|q_{t-1}}(t)$ for the chosen arm in the last time slot using Eq. (19). According the Hoeffding's inequality, [42] gives an upper bound of the estimated reward $r_{a_{t-1}^*|q_{t-1}}(t)$ as follows,

$$H_{a_{t-1}^*|q_{t-1}} = \frac{R_{a_{t-1}^*|q_{t-1}}}{N_{a_{t-1}^*|q_{t-1}}} + \sqrt{\frac{2\ln t}{N_{a_{t-1}^*|q_{t-1}}}}, \tag{21}$$

where $N_{a_{t-1}^*|q_{t-1}}$ stands for the number of times the arm a^* has been selected by time slot $t-1$, and $R_{a_{t-1}^*|q_{t-1}}$ means the total reward of choosing a^* over the past $t-1$ time slots. $R_{a_{t-1}^*|q_{t-1}}$ is calculated by

$$R_{a_{t-1}^*|q_{t-1}} = \sum_{x \in \{y | a_y = a_{t-1}^* \cap q_y = q_{t-1}, y \leq t-1\}} r_{a_x|q_x} \tag{22}$$

To determine the action a_t^* given the new context $\hat{\mathbf{x}}_t^s \to q_t$, the UCB algorithm will compare the reward of the chosen arms with that of non-chosen arms to solve the exploitation vs. exploration dilemma. During the exploitation phase, the UCB algorithm favors actions that have shown higher upper bound $H_{a|q}$ of the estimated reward. By exploiting the currently known best actions, the algorithm aims to maximize the expected reward and make optimal decisions based on the existing knowledge. On the other hand, during the exploration phase, if an action has not been tried very often, or not at all, then $N_{a|q}$ will be small. Consequently the second term of (21) will be large, making this action more likely to be selected. This allows the algorithm to collect more data and gain a better understanding of the potential rewards associated with different actions.

With UCB, a_t^* is selected according to

$$a_t^* = \arg\max_{a \in \cap \hat{\mathcal{A}}_t} \{H_{a|q_t}\}. \tag{23}$$

5 Evaluations

In this section, we examine the performance of QC-MAB by comparing it to the two state-of-the-art algorithms below.

- Peekaboo [12]: it is a novel CMAB-based MPQUIC scheduler. It strives for learning the dynamic characteristics of the heterogeneous path including path RTT and loss rate, and then makes decisions about whether to schedule packets into the path with undesirable conditions immediately or wait for the availability of the path with good conditions.
- LEAP [15]: similar to us, LEAP considers three QoS metrics, i.e., delay, reliability and goodput. It employs FEC to trade off among the three metrics,

This article has been accepted for publication in IEEE Transactions on Mobile Computing. 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/TMC.2024.3507051

and uses a bandwidth-aware multipath congestion control algorithm to avoid overshooting issues.

- MARS [43]: a recent DRL-based scheduling algorithm for video streaming over MPQUIC. MARS takes into account out-of-order queue size and different QoS metrics to realize adaptive scheduling optimization.

5.1 Evaluation Methodology

5.1.1 Testbed Construction

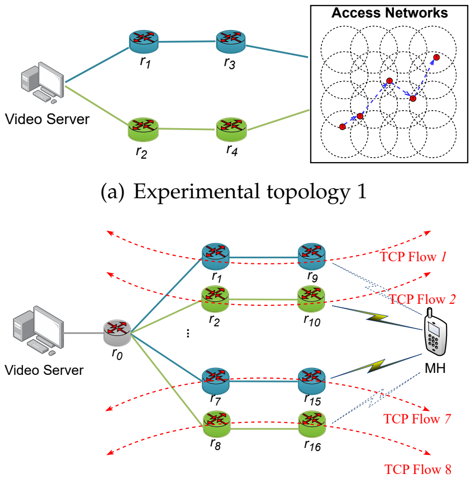
To start, we extended the existing QUIC module in network simulator 3 (ns-3) [44] to support MPQUIC, following the guidelines specified in the Internet Engineering Task Force (IETF) drafts [45] on MPQUIC. The publicly available version of our MPQUIC source code can be accessed at [46]. Using this platform, we implemented our proposed solution and incorporated two existing algorithms for comparative analysis. Additionally, we referred to the DASH source code at [47] and made modifications to the MPQUIC source code to enable DASH to operate over MPQUIC.

In order to replicate network bottleneck scenarios that can occur at any point along the E2E path, including access networks and backbone networks, we created two network topologies, as depicted in Fig. 4. In Fig. 4(a), we evaluated a situation where the end user moves across various access networks with different conditions, some of which may become bottlenecks. Fig. 4(b) represents two cases: 1) when the end user remains stationary while multiple background flows compete for backbone resources, causing significant variations in data rate, link delay, and packet drop rate, indicating the backbone network as a potential bottleneck, and 2) when the end user remains stationary and there are no background flows along the E2E paths.

Both Fig. 4(a) and Fig. 4(b) depict that an MH is equipped with a WiFi interface and a 5G interface, allowing for the establishment of two subflows between the MH and the video server. In Fig. 4(a), the backbone network comprises a 2×2 node configuration, with circular markers representing the communication range of the AP nodes. We employed the *WiFiHelper* or *NrHelper* tools to configure the access links between the MH and the APs, while the *PointToPointHelper* tool was used to set up the wired links. In Fig. 4(b), the backbone network consists of an 8×2 node configuration, with the rightmost column of nodes acting as APs to connect the MH. The solid flash between an AP and the MH represents the currently utilized access link, while the dashed one represents a candidate link.

5.1.2 Video Application Settings

In each simulation, we employed the Dynamic Adaptive Streaming over HTTP (DASH) protocol at the mobile host (MH) side to retrieve video segments from the video server. The video segments used in our simulations were provided by the authors of [48] and were extracted from the movie "Sintel." This dataset comprises 84 video segments, with video bitrates ranging from 0.088 Mbps to 20.5 Mbps, encompassing eight quality increments. The selection of bitrates for different representations was designed to achieve a roughly linear increase in video quality, as measured by the Peak Signal-to-Noise Ratio (PSNR). Therefore, we can utilize the representation index as an indicator of video quality.



(a) Experimental topology 1

(b) Experimental topology 2

Figure 4: Experimental topologies.

In practical video data protection scenarios, FEC codes are often employed at the bit level, group of pictures (GoP) level, or frame level [49]. In our paper, we adopt a frame-level FEC approach at the MPQUIC layer to allocate FEC redundancy for each coding layer within the frame. In accordance with the settings in [49], our SVC configuration includes three layers (one base layer and two enhancement layers), with a FEC redundancy of 30% for each layer. To ensure acceptable video quality, we impose a constraint on the per-packet elapsed delay, limiting it to less than 150 ms. This constraint aligns with previous studies [50], [51] that suggest a one-way delay not exceeding 150 ms to achieve excellent video quality. Consequently, the video player displays one frame every ΔT seconds, where ΔT is set to 0.15. If the base layer of a frame fails to decode successfully for display, a playback interruption occurs. Let n_0 represent the number of playback interruptions, and n_t denote the total number of frames. The interruption ratio (IR) is then defined as IR = n_0/n_t, which serves as a critical measure of video user experience.

For all scenarios examined, we measure four key performance metrics: interruption ratio (IR), mean video quality, E2E delay for all BL packets, and aggregated goodput. These metrics enable us to evaluate the impact of our proposed solution and compare it with existing algorithms in terms of video playback experience, video quality, delay, and overall network throughput.

5.2 User Movement in Ultra-Dense Networks

5.2.1 Experimental Settings

In this scenario, we utilize the experimental topology depicted in Fig. 4(a). Inspired by previous studies [52], [53], we configure the mobile access networks as follows: the APs are evenly distributed in a lattice grid topology, with neighboring APs placed at a distance of 20 meters. The MH initiates its movement from the lower-left corner and randomly moves in a specified direction at a velocity of

10 m/s. The transmission power level (P) of the APs is randomly selected within the range of 100 mW to 200 mW. The path loss exponent (e) is set to 4. We assume no interference and set the noise power (σ) to -180 dBm. The allocated bandwidth (W) of the APs to the MH ranges from 1 MHz to 2 MHz. Based on insights from prior studies [54], we set the MH's handoff threshold to -90 dBm. For the wired links represented by solid lines in Fig. 4(a), we initialize the link data rate to 100 Mbps, link packet error rate to 0.01%, and the link delay to a random value ranging from 5 ms to 20 ms, aligning with the settings in [1].

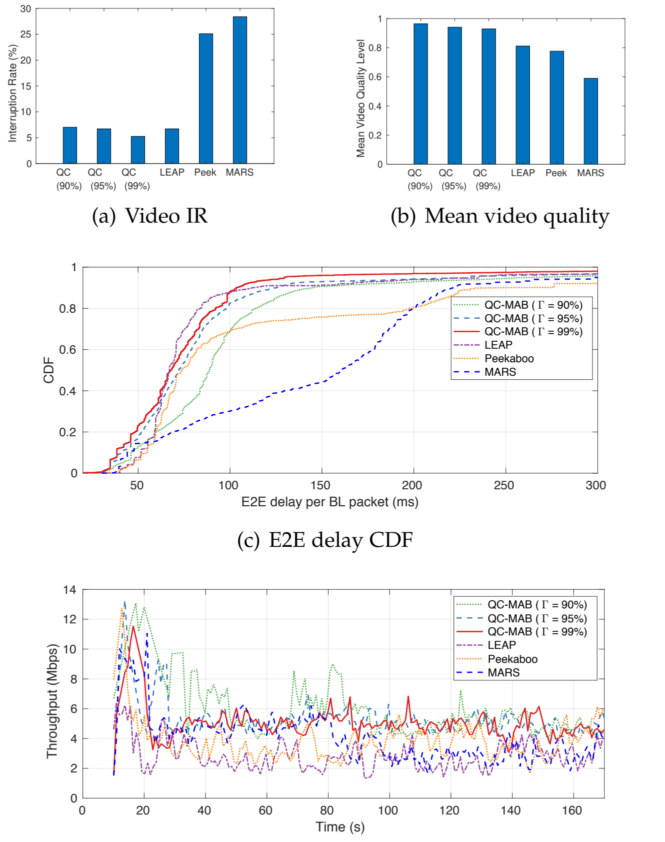
Regarding the QC-MAB algorithm, we define several thresholds to partition the context space, as discussed in Section 4.6. Through a series of tests and based on the aforementioned settings, we set the bandwidth threshold to 5 Mbps and 10 Mbps, the OWD threshold to 90 ms and 150 ms, and the error rate threshold to 0.01% and 0.1%. It is important to note that the selection of partition thresholds is highly dependent on the specific scenario, and intelligently tuning these thresholds to adapt to various scenarios remains an open issue. We set the coefficient α in Equation (17) to $\frac{1}{8}$, as suggested in [55], and the coefficient β in Equation (18) to 0.6, as indicated in [56]. Both QC-MAB and LEAP utilize the QoS requirements as input, and for consistency with LEAP, we set the deadline ΔT of BL packets to 150 ms. The reliability requirement (Γ) for QC-MAB is set at three levels: 90%, 95%, and 99%, while LEAP maintains a reliability requirement of 99%.

5.2.2 IR and E2E Delay

Fig. 5 presents the QoS performance under the aforementioned settings. In terms of IR performance (Fig. 5(a)), we observe that both Peekaboo and MARS exhibit an interruption ratio of more than 25%, while QC-MAB and LEAP maintain a ratio of 5%. The higher IR observed in Peekaboo is attributed to its focus on QoS improvement rather than QoS guarantee. Peekaboo does not penalize the system when BL packets fail to arrive on time, resulting in a higher IR. The highest IR record in MARS reveals that DRL training in mobile environments is too slow to adapt to the dynamics. This observation is further supported by the results of E2E delay in Fig. 5(c).

Analyzing Fig. 5(c), we observe that QC-MAB with a larger Γ value achieves lower E2E delay, indicating that the actions taken by QC-MAB are indeed driven by QoS requirements. Similarly, LEAP also considers the stringent reliability requirements and employs congestion control and FEC strategies to meet those requirements. However, the probability of achieving an E2E delay below 150 ms is 91.1% for LEAP, whereas QC-MAB achieves 95.9% with a reliability requirement of $\Gamma = 99\%$.

When it comes to reliability, MARS performs the worst at 44%. This can be attributed to the increasing BER as the MH moves away from connected APs. While LEAP faces a similar challenge, it utilizes FEC techniques to recover BL packets without retransmission, resulting in better E2E delay performance compared to MARS. In QC-MAB, in addition to employing FEC techniques, the intelligent network selection strategy allows it to proactively explore other suitable APs based on wireless link contexts, effectively mitigating the negative impact of mobility.



(a) Video IR

(b) Mean video quality

(c) E2E delay CDF

(d) Aggregated goodput

Figure 5: QoS performance when the access networks or user is in motion.

Overall, the experimental results demonstrate that QC-MAB outperforms Peekaboo and MARS and achieves comparable performance to LEAP in terms of QoS metrics such as IR and E2E delay. The intelligent network selection and proactive AP switching capabilities of QC-MAB enable it to adapt to changing network conditions and ensure reliable and high-quality video streaming in ultra-dense network environments.

5.2.3 Mean Video Quality and Goodput

To achieve high video quality beyond playback smoothness, QC-MAB aims to explore better access networks that can transmit more EL packets without violating the delay requirement of BL packets. Fig. 5(b) illustrates the mean video quality results, showing that QC-MAB achieves higher mean quality compared to the benchmark algorithms. Additionally, there is a tradeoff between reliability and mean video quality in mobile scenarios, where higher reliability requirements lead to lower mean quality.

The video quality heavily depends on the aggregated goodput over multiple paths. Fig. 5(d) presents the changes in goodput over time for different algorithms in the context of mobility. QC-MAB achieves up to three times higher goodput than LEAP and Peekaboo when Γ is set to 90% and 95%. The goodput decreases when a higher reliability requirement, such as 99%, is imposed, aligning with the behavior of lower mean quality with higher reliability. How-

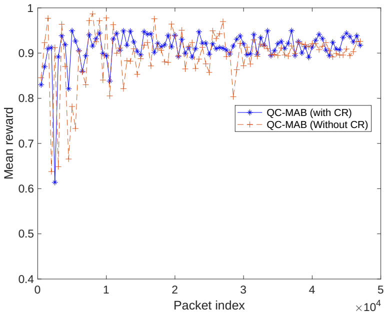


Figure 6: The effectiveness of the CR component.

ever, even with a reliability requirement of 99%, QC-MAB still improves the goodput compared to LEAP and Peekaboo due to the exploration of diverse APs during movement. The goodput of MARS maintains at 6 Mbps in the first 80 seconds and then drops down to 2.5 Mbps.

5.2.4 Convergence and Complexity Analysis

The key component that improves the convergence rate in QC-MAB is the CR, which includes context dimension reduction and context noise removal. Fig. 6 compares the mean reward performance of QC-MAB with and without CR. Without CR, the mean reward is lower, and the convergence rate is slower. CR reduces the dimensionality of the state space while also removing noise, resulting in faster and more accurate relationship discovery between the state and QoS performance. Without CR, inappropriate choices may be made, leading to larger variations in instantaneous reward.

To gain more insights about the overhead of running our algorithm, we measured the average elapsed time from when we started running to when the decision was made. Before using CR, the averaging running time is 0.6 microseconds, while it is reduced to 0.03 microseconds after CR is enabled. Therefore, our algorithm is well-suited for delay-sensitive applications.

5.2.5 Impact of Access Network Density

The density of the access network determines the number of visible networks to the end user, which affects the actions set of QC-MAB. In this subsection, we explore the impact of access network density by adjusting the distances between neighboring APs.

The experiments reveal that LEAP and Peekaboo do not proactively select networks, resulting in minimal changes in their IR performance as the distances change. Therefore, we analyze the IR performance with QC-MAB under varied AP distances, as shown in Fig. 7. As the AP distance increases, the IR with QC-MAB initially decreases until a certain threshold and then starts to increase. The analysis of AP density demonstrates the exploration and exploitation dilemma: when more access networks are visible to the end user, QC-MAB has a higher chance of finding better networks to exploit after several rounds of exploration. However, when the density exceeds a certain threshold, continuously exploring new choices may negatively impact the IR performance, as QC-MAB could explore networks with poor conditions.

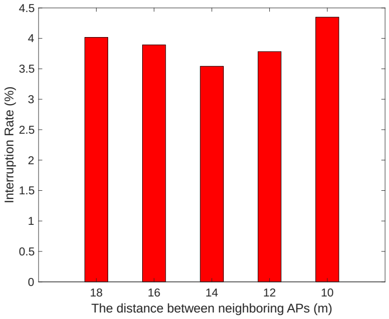


Figure 7: IR performance under the varied density of network coverage.

Overall, the experimental results highlight the advantages of QC-MAB in terms of mean video quality, goodput, convergence rate, and its ability to adapt to varying network densities in ultra-dense environments.

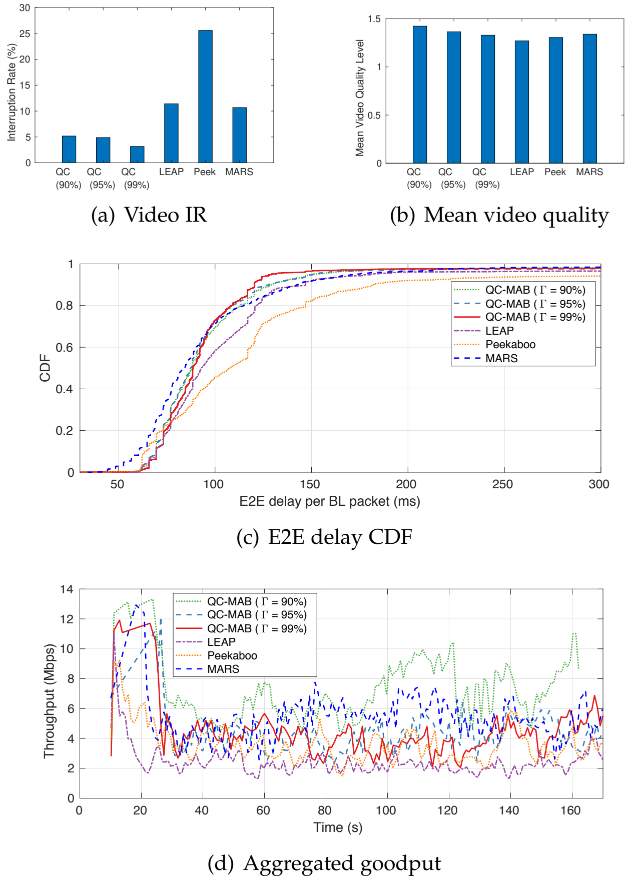
5.3 Impact of Background Traffic in Backbone Networks

Traditional backbone networks are typically high-capacity and stable. However, in the next-generation network, this may not always be the case. For example, satellite networks can serve as backbone networks, but their inter/intra satellite links are not as stable as terrestrial networks, making them potential bottleneck networks due to satellite movement. To simulate this situation, we constructed the topology shown in Fig. 4(b). In this topology, all link settings regarding the delay and error rate remain unchanged, except that several background TCP flows share the wired links, which have a lower data rate compared to the rightmost access links. Specifically, the bandwidth of access links is set to 30 Mbps, while that of the wired links is 20 Mbps [57]. Each background TCP flow handles burst traffic within the same lifespan as the MPQUIC flows.

5.3.1 IR and E2E Delay

As shown in Fig. 8(c), 82.19%, 91.98%, and 92.03% of packets meet the 150 ms deadline when using Peekaboo, LEAP, and MARS, respectively, while QC-MAB achieves 96.7% reliability with $\Gamma = 99\%$. In Peekaboo, the dynamic bottleneck bandwidth is not taken into account, leading to inappropriate scheduling decisions. LEAP heavily relies on bottleneck bandwidth estimation to determine the cwnd value. However, this estimation is inaccurate when the bandwidth frequently changes over time. MARS can adapt to the bandwidth dynamics by using DRL, however, the reward design is not driven by the reliability threshold. In QC-MAB, we emphasize the importance of the bottleneck bandwidth by integrating it into the context space and using an AKF to remove bandwidth noise. This approach ensures that the connection between bandwidth and QoS reward is accurately revealed, resulting in high reliability. According to Fig. 8(a), QC-MAB reduces video IR to around 3.1%, while LEAP and MARS achieve the IR of 11.4%. Peekaboo performs worse, with the IR of 25.11%.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. 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/TMC.2024.3507051



(a) Video IR

(b) Mean video quality

(c) E2E delay CDF

(d) Aggregated goodput

Figure 8: QoS performance when the backbone network experiences large variations and becomes the bottleneck.

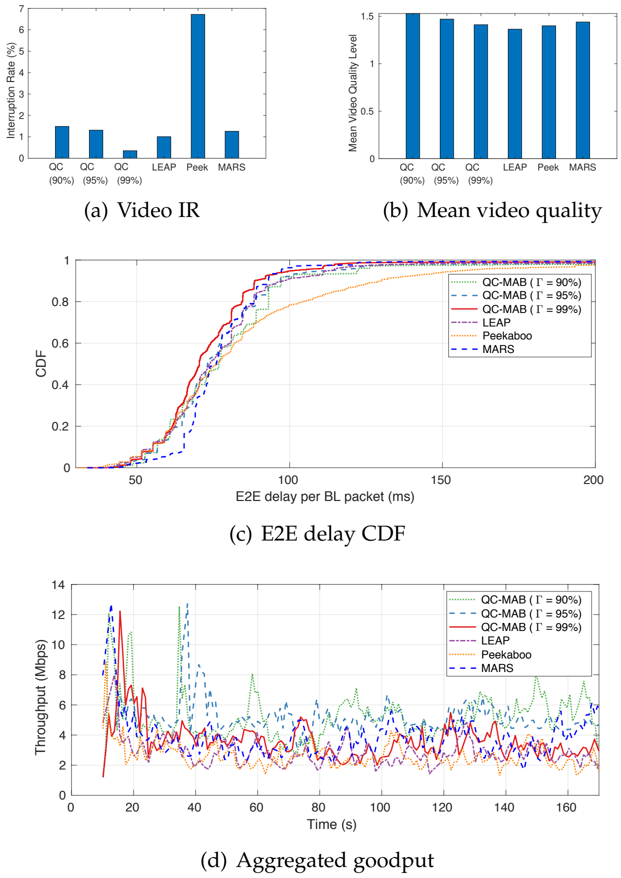
5.3.2 Mean Video Quality and Goodput

From Fig. 8(d), we observe that both Peekaboo and LEAP do not have a sharp increase in cwnd at the beginning as shown in Fig. 9(d). This is caused by the lack of awareness of bandwidth variations in Peekaboo and LEAP, resulting in overshooting issues. In comparison, MARS achieves a better goodput score in non-mobile environments. QC-MAB with different Γ values also provides higher goodput than Peekaboo and LEAP because it utilizes the UCB algorithm to balance exploration and exploitation. When high variations in routes are detected, QC-MAB can proactively switch to other access points for higher rewards. In comparison to Peekaboo, LEAP achieves lower goodput due to the packet redundancy generated by FEC.

Fig. 8(b) illustrates the mean video quality performances, which align with the goodput behaviors. QC-MAB overall improves the mean video quality compared to LEAP and Peekaboo, while LEAP achieves the lowest video quality level.

5.4 Without Background Traffic

In this subsection, we examine how QC-MAB performs in a scenario without background traffic competing for bottleneck resources, based on the topology shown in Fig. 4(b). In this subsection, except for the absence of background TCP flows, all other experimental settings remain the same as in Section 5.3.



(a) Video IR

(b) Mean video quality

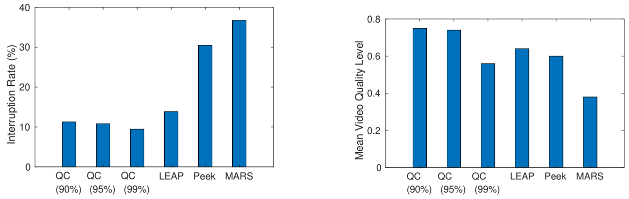
(c) E2E delay CDF

(d) Aggregated goodput

Figure 9: QoS performance when the backbone network and access networks are relatively stable.

Compared to the IR performances in the previous two scenarios with high dynamics, the overall IR values in this scenario are significantly reduced. Specifically, the IR values with MARS, Peekaboo, LEAP, and QC-MAB are reduced from an average of 28% to 1%, from 25% to 7%, from 6% to 1%, and from 5% to 0.5%, respectively. The differences in IR performance across different scenarios highlight that QC-MAB and LEAP outperform Peekaboo in terms of QoS guarantee. On the other hand, MARS exhibits a significant difference w.r.t. the IR scores in mobile environments and non-mobile environments. This reveals the limitation of MARS in dealing with mobility issues. As shown in Fig. 9(c), QC-MAB, MARS, and LEAP achieve similar E2E delay guarantee performances, with over 99% of BL packets arriving within 150 ms. Peekaboo delivers 96% of BL packets within 150 ms.

Fig. 9(b) presents the mean video quality comparison, showing that while LEAP maintains similar IR values to QC-MAB, it has the lowest mean video quality level. This phenomenon can be explained by the goodput results shown in Fig. 9(d). Compared to others, LEAP exhibits the lowest goodput throughout the streaming process. As mentioned earlier, LEAP often compromises bandwidth to guarantee the shortest delay for the most important video frames.



(a) Video IR

(b) Mean video quality

Figure 10: QoS performance comparison for 360-degree videos in mobile networks.

5.5 Validations over 360-degree Videos

360-degree video streaming is a crucial enabler for killer applications in 6G such as virtual reality (VR) and augmented reality (AR), as it captures scenes in all directions (i.e., panoramic views) and provides an immersive experience for users. Compared to planar video streaming, 360-degree video streaming is more bandwidth-intensive and delay-sensitive. To gain more insights into how the state-of-the-art algorithms behave to streaming 360-degree videos in mobile environments, in this subsection, we briefly showcase the performances of IR and mean quality when using different algorithms to transmit 360-degree videos.

In our experiments, we adopt a video dataset from [58] consisting of ten 360-degree videos, each with a duration of 60 seconds. Each 60-second video comprises 1800 frames, with each frame lasting approximately 33 milliseconds. Each group of pictures (GoP) comprises 50 frames, including 1 BL frame and 49 EL frames. The size of each frame varies from 500 KB to 10 MB. In general, BL frames have larger file sizes compared to EL frames due to their comprehensive encoding. Each frame in our experiments is segmented sinto multiple packets with the same size of 1500 bytes. The deadline for BL packets is set to 100 ms.

Fig. 10 appears a similar trend as Fig. 5. Specifically, Fig. 10(a) shows that Peekaboo and MARS exhibit a higher interruption ratio compared to LEAP and QC-MAB. Fig. 10(b) demonstrates that QC-MAB achieves the greatest video quality while rendering a lower IR value compared to benchmark algorithms. Therefore, QC-MAB still improves against benchmark algorithms in mobile environments when dealing with 360-degree videos.

However, compared to the IR performance of QC-MAB over planar videos, the counterpart of QC-MAB over 360-degree videos is doubly increased. One reason is that the deadline requirement for BL packets of 360 videos is tighter, i.e., 100 ms. In addition to that, 360-degree videos have more salient features that affect the QoS metrics. Therefore, the specific study on the QoS guarantee for 360-degree video streaming will be our future research focus.

6 Conclusions and Future Work

In this paper, we develop a novel learning-based framework, named QC-MAB, to solve the QoS guarantee issue for multipath video streaming in mobile networks. As the QC-MAB incorporates both wireless link characteristics changes and E2E path conditions change in its context space design, in conjunction with context and arm dimension reduction, it can discover the underlying relationship between network context and QoS performance without a lengthy training process. Given a particular context and pre-defined QoS requirements, QC-MAB can therefore make a reasonable decision with regard to access network selection and FEC coding to trade off delay, reliability, and goodput. Extensive experiments based on ns-3 demonstrate that QC-MAB not only provides an ultra-low video interruption ratio but also high goodput in heterogeneous stationary networks or dynamic mobile systems.

The process of proactively switching from one AP to another is complex in practice. It involves synchronization, association, and service migration, all of which present challenges in implementing our algorithm.

Additionally, frequent switching can introduce significant overhead. In future research, we plan to incorporate realistic handoff procedures to balance the benefits of exploring new access networks with the overhead of network switching. Furthermore, utilizing adaptive coding rates offers more opportunities for exploration, but it also presents challenges due to the vast choice space associated with coding rate options.

References

[1] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, “Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2345–2361, 2015.

[2] W. Yang, S. Shu, L. Cai, and J. Pan, “MM-QUIC: Mobility-aware multipath QUIC for satellite networks,” in *17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 608–615.

[3] W. Yang, L. Cai, S. Shu, and J. Pan, “Scheduler design for mobility-aware multipath QUIC,” in *Proceedings IEEE GLOBECOM*, 2022, pp. 2849–2854.

[4] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan, “MP-DASH: Adaptive video streaming over preference-aware multipath,” in *Proceedings of the 12th International on Conference on Emerging Networking Experiments and Technologies*, 2016, pp. 129–143.

[5] T. Stockhammer, “Dynamic adaptive streaming over HTTP–standards and design principles,” in *Proceedings of the 2nd ACM Conference on Multimedia Systems*, 2011, pp. 133–144.

[6] M. Xing, S. Xiang, and L. Cai, “A real-time adaptive algorithm for video streaming over multiple wireless access networks,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 795–805, 2014.

[7] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with Pensieve,” in *Proceedings of ACM Conference on Special Interest Group on Data Communication*, 2017, pp. 197–210.

[8] H. Zhang, W. Li, S. Gao, X. Wang, and B. Ye, “ReLeS: A neural adaptive multipath scheduler based on deep reinforcement learning,” in *Proceedings IEEE INFOCOM*, 2019, pp. 1648–1656.

[9] Y. Guan, Y. Zhang, B. Wang, K. Bian, X. Xiong, and L. Song, “PERM: Neural adaptive video streaming with multi-path transmission,” in *Proceedings IEEE INFOCOM*, 2020, pp. 1103–1112.

[10] A. Sepahi, L. Cai, W. Yang, and J. Pan, “Meta-DAMS: Delay-aware multipath scheduler using hybrid meta reinforcement learning,” in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023, pp. 1–5.

[11] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, “DRL360: 360-degree video streaming with deep reinforcement learning,” in *Proceedings IEEE INFOCOM*, 2019, pp. 1252–1260.

[12] 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.

[13] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, “FMTCP: A fountain code-based multipath transmission control protocol,” *IEEE/ACM Transactions On Networking*, vol. 23, no. 2, pp. 465–478, 2014.

[14] M. Li, A. Lukyanenko, and Y. Cui, "Network coding based multi-path TCP," in *IEEE INFOCOM Workshops*, 2012, pp. 25–30.

[15] F. Chiariotti, S. Kucera, A. Zanella, and H. Claussen, "Analysis and design of a latency control protocol for multi-path data delivery with pre-defined QoS guarantees," *IEEE/ACM Transactions On Networking*, vol. 27, no. 3, pp. 1165–1178, 2019.

[16] L. Cai, J. Pan, W. Yang, X. Ren, and X. Shen, "Self-evolving and transformative (set) protocol architecture for 6G," *IEEE Wireless Communications*, 2022.

[17] J. Wu, C. Yuen, M. Wang, and J. Chen, "Content-aware concurrent multipath transfer for high-definition video streaming over heterogeneous wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 3, pp. 710–723, 2015.

[18] "H.265." [Online]. Available: https://www.itu.int/rec/T-REC-H.265-202108-I/en

[19] L. Cai, Y. Luo, S. Xiang, and J. Pan, "Scalable modulation for scalable wireless videocast," in *Proceedings IEEE INFOCOM*, 2010, pp. 1–5.

[20] X. Zhu, R. Pan, M. S. Prabhu, N. Dukkipati, V. Subramanian, and F. Bonomi, "Layered internet video adaptation (LIVA): Network-assisted bandwidth sharing and transient loss protection for video streaming," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 720–732, 2011.

[21] 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, pp. 187–198.

[22] 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, pp. 325–338.

[23] G. Lv, Q. Wu, W. Wang, Z. Li, and G. Xie, "Lumos: Towards better video streaming QoE through accurate throughput prediction," in *Proceedings IEEE INFOCOM*, 2022, pp. 650–659.

[24] W. Yang, P. Dong, L. Cai, and W. Tang, "Loss-aware throughput estimation scheduler for multi-path TCP in heterogeneous wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 3336–3349, 2021.

[25] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and evaluation," in *Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies*, 2017, pp. 160–166.

[26] E. Dong, M. Xu, X. Fu, and Y. Cao, "A loss aware MPTCP scheduler for highly lossy networks," *Computer Networks*, vol. 157, pp. 146–158, 2019.

[27] J. Wang, Y. Gao, and C. Xu, "A multipath QUIC scheduler for mobile HTTP/2," in *Proceedings of the 3rd Asia-Pacific Workshop on Networking*, 2019, pp. 43–49.

[28] V. A. Vu and B. Walker, "On the latency of multipath-QUIC in real-time applications," in *16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2020, pp. 1–7.

[29] A. Rabitsch, P. Hurtig, and A. Brunstrom, "A stream-aware multipath QUIC scheduler for heterogeneous paths," in *Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC*, 2018, pp. 29–35.

[30] J. Wu, B. Cheng, C. Yuen, Y. Shang, and J. Chen, "Distortion-aware concurrent multipath transfer for mobile video streaming in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 688–701, 2014.

[31] Y. E. Guo, A. Nikravesh, Z. M. Mao, F. Qian, and S. Sen, "Accelerating multipath transport through balanced subflow completion," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 141–153.

[32] X. Zuo, Y. Cui, X. Wang, and J. Yang, "Deadline-aware multipath transmission for streaming blocks," in *Proceedings IEEE INFOCOM*, 2022, pp. 2178–2187.

[33] A. Alzadjali, F. Esposito, and J. Deogun, "A contextual bi-armed bandit approach for MPTCP path management in heterogeneous LTE and WiFi edge networks," in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, 2020, pp. 307–316.

[34] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: A randomized experiment in video streaming," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020, pp. 495–511.

[35] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.

[36] H. W. Oh, I. T. Han, K. R. Park, and S. H. Kim, "An enhanced multi-path scheme for QoS guarantee in wireless sensor network," in *IEEE International Symposium on Consumer Electronics*, 2007, pp. 1–5.

[37] Y. Tang, S. Agrawal, and Y. Faenza, "Reinforcement learning for integer programming: Learning to cut," in *International conference on machine learning*. PMLR, 2020, pp. 9367–9376.

[38] J. Wu, C. Yuen, B. Cheng, Y. Shang, and J. Chen, "Goodput-aware load distribution for real-time traffic over multipath networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2286–2299, 2014.

[39] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[40] M. R. Rajamani and J. B. Rawlings, "Estimation of the disturbance structure from data using semidefinite programming and optimal weighting," *Automatica*, vol. 45, no. 1, pp. 142–148, 2009.

[41] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002.

[42] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[43] X. Han, B. Han, J. Li, and C. Song, "Multi-agent DRL-based multipath scheduling for video streaming with QUIC," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 20, no. 7, pp. 1–23, 2024.

[44] "quic-ns-3." [Online]. Available: https://github.com/signetlabdei/quic-ns-3

[45] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-multipath-03, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/03/

[46] "MPQUIC-ns3," https://github.com/ssjShirley/mpquic-ns3, accessed: 2023-04-20.

[47] "DASH-NS3." [Online]. Available: https://github.com/haraldott/dash

[48] H. Ott, K. Miller, and A. Wolisz, "Simulation framework for http-based adaptive streaming applications," in *Proceedings of the Workshop on ns-3*, 2017, pp. 95–102.

[49] X. Zhang, X. Zhang, Z. Fu, B. Yu, and S. Fowler, "Joint distortion estimation and layer selection of unequal error protection for svc video transmission over fso networks," in *IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019, pp. 726–733.

[50] W. Song and W. Zhuang, "Performance analysis of probabilistic multipath transmission of video streaming traffic over multi-radio wireless devices," *IEEE Transactions on Wireless Communications*, vol. 11, no. 4, pp. 1554–1564, 2012.

[51] J. Wu, B. Cheng, C. Yuen, N.-M. Cheung, and J. Chen, "Trading delay for distortion in one-way video communication over the internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 711–723, 2015.

[52] S. Aeron and V. Saligrama, "Wireless ad hoc networks: Strategies and scaling laws for the fixed snr regime," *IEEE Transactions on Information Theory*, vol. 53, no. 6, pp. 2044–2059, 2007.

[53] X. Yang, H. Lin, Z. Li, F. Qian, X. Li, Z. He, X. Wu, X. Wang, Y. Liu, Z. Liao *et al.*, "Mobile access bandwidth in practice: Measurement, analysis, and implications," in *Proceedings of the ACM SIGCOMM Conference*, 2022, pp. 114–128.

[54] K. Yang, I. Gondal, B. Qiu, and L. S. Dooley, "Combined SINR based vertical handoff algorithm for next generation heterogeneous wireless networks," in *Proceedings IEEE GLOBECOM*, 2007, pp. 4483–4487.

[55] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.

[56] R. Wang, M. Valla, M. Sanadidi, and M. Gerla, "Adaptive bandwidth share estimation in TCP Westwood," in *Proceedings IEEE GLOBECOM*, vol. 3, 2002, pp. 2604–2608.

[57] S. Ferlin, Ö. Alay, T. Dreibholz, D. A. Hayes, and M. Welzl, "Revisiting congestion control for multipath TCP with shared

bottleneck detection," in *Proceedings IEEE INFOCOM*, 2016, pp. 1–9.

[58] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360 video viewing dataset in head-mounted virtual reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017, pp. 211–216.