COREL: Constrained Reinforcement Learning for Video Streaming ABR Algorithm Design over mmWave 5G

Xinyue Hu
University of Minnesota - Twin Cities

Arnob Ghosh
New Jersey Institute of Technology

Xin Liu
University of California - Davis

Zhi-Li Zhang
University of Minnesota - Twin Cities

Ness Shroff
The Ohio State University

***Abstract*—The adaptive bitrate selection (ABR) mechanism, which decides the bitrate for each video chunk is an important part of video streaming. There has been significant interest in developing Reinforcement-Learning (RL) based ABR algorithms because of their ability to learn efficient bitrate actions based on past data and their demonstrated improvements over wired, 3G and 4G networks. However, the Quality of Experience (QoE), especially video stall time, of state-of-the-art ABR algorithms including the RL-based approaches falls short of expectations over commercial mmWave 5G networks, due to widely and wildly fluctuating throughput. These algorithms find optimal policies for a multi-objective unconstrained problem where the policies inherently depend on the predefined weight parameters of the multiple objectives (e.g., bitrate maximization, stall-time minimization). Our empirical evaluation suggests that such a policy cannot adequately adapt to the high variations of 5G throughput, resulting in long stall times.**

To address these issues, we formulate the ABR selection problem as a constrained Markov Decision Process where the objective is to maximize the QoE subject to a stall-time constraint. The strength of this formulation is that it helps mitigate the stall time while maintaining high bitrates. We propose COREL, a primal-dual actor-critic RL algorithm, which incorporates an additional critic network to estimate stall time compared to existing RL-based approaches and can tune the optimal dual variable or weight to guide the policy towards minimizing stall time. Our experiment results across various commercial mmWave 5G traces reveal that COREL reduces the average stall time by a factor of 4 and the 95th percentile by a factor of 2.

I. Introduction

Video traffic accounts for about 70% of mobile data traffic and is expected to rise to 80% by 2028 [1]. The emergence of 5G networks, especially millimeter-wave (mmWave) 5G, with their ultra-high bandwidth capabilities, have positioned them as key enablers for bandwidth-intensive applications, such as 4K/8K video, 360 video, and volumetric video streaming.

Video streaming today typically involves dividing the video into smaller chunks, each encoded at various bitrates. An important functionality of video streaming is adaptive bitrate selection (ABR), which decides the appropriate bitrate for each video chunk. The goal of ABR algorithms is to find a balance between increasing video quality and avoiding playback stalls. Designing an optimal ABR algorithm with hand-tuned heuristics is difficult, due to hard-to-model network dynamics and hard-to-balance video QoE objectives. Facing these challenges, recent ABR algorithms [2]–[4] leverage statistical and machine-learning techniques to make bitrate selections based on historical data about throughput, buffer occupancy, and download time. Among these approaches, reinforcement learning (RL) based ABR algorithms (*e.g.,* Pensieve [3]) have shown promise by optimizing policies utilizing the performance of past decisions to enable the discovery of superior policies compared to algorithms that use fixed heuristics or inaccurate system models.

However, the QoE performance, especially video stall time, of state-of-the-art ABR algorithms over commercial mmWave 5G networks falls short of expectation, as shown in a recent measurement study [5]. Although commercial mmWave 5G can indeed offer ultra-high bandwidth (*e.g.,* up to 2 Gbps) [6], [7], the wild fluctuations in 5G throughput, caused by factors such as the directional nature of mmWave signals and environmental conditions, pose significant challenges for existing ABR algorithms to achieve satisfactory QoE performance, especially in mobile scenarios. Interestingly, Pensieve, which outperforms other ABR algorithms in 3G and 4G networks, exhibits the highest bitrate as well as the *highest* stall time under 5G. In summary, the primary concern for video streaming over mmWave 5G lies in the high video stall time.

To mitigate video stall time caused by the volatile nature of 5G throughput while effectively utilizing the high bandwidth, this paper proposes to employ constrained reinforcement learning to design an ABR algorithm for mmWave 5G. In particular, we formulate the optimization of the ABR algorithm as a *constrained* Markov decision process (CMDP). Unlike existing ABR algorithms [2], [3], [8] that directly optimize a pre-defined weighted multiple-objective QoE metric that is agnostic to the network environment, our approach focuses on optimizing a single objective (*i.e.,* maximizing bitrate) while maintaining a controlled level of performance degradation in other objectives (*i.e.,* minimizing stall time to satisfy a constraint). To solve the proposed CMDP, we propose a primal-dual RL algorithm, called COREL. In particular, we tune the dual variable (*i.e.,* the weight parameter for the stall-time) based on the stall time estimated by the critic network, as well as tune the policy parameter based on the performance on the corresponding QoE simultaneously. This approach allows us to find the policy that aims to satisfy the stall time constraint while maintaining overall good throughput utilization.

We have trained and tested COREL on various commercial mmWave 5G network conditions. To obtain extensive network

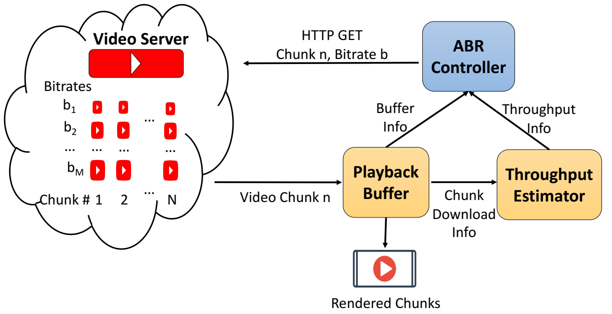


Fig. 1: An overview of HTTP adaptive video streaming.

traces, we augment the existing Lumos5G dataset [7], which consists of 20 hours of mmWave 5G traces, by collecting an additional 169 hours of mmWave 5G traces. Compared to state-of-the-art ABR algorithms, COREL achieves an outstanding balance between high bitrate and low stall time. It outperforms the best existing schemes with a reduction in average stall time ranging from 40.63% to 75.27% and a reduction in tail stall time at 95-percentile from 25.24% to 58.44%, at a *similar* bitrate level. In summary, COREL reduces the stall-time significantly while maintaining the high bitrate needed to sustain video streaming services.

II. Background & Related Work

A. Basics of Adaptive Video Streaming

HTTP-based adaptive video streaming has become the most prevalent method for video on demand (VoD) streaming. Fig. 1 illustrates the process of VoD streaming over HTTP. The video server needs to serve multiple clients with diverse and unpredictable connection performance. Therefore, the server divides the video into chunks, usually 2-6 seconds long, and encodes each chunk independently at a few different bitrates. On the client side, the client requests chunks one by one and employs an adaptive bitrate (ABR) algorithm to select a suitable bitrate for each chunk. The ABR algorithms make decisions based on recent experience and predictions about the future (*e.g.,* playback buffer occupancy, throughput estimation). The goal of an ABR algorithm is to reduce the stall time, maximize the quality of chunks, and minimize variation in quality over time. After chunks are downloaded and stored in the playback buffer, they are played back to the client. Note that the playback of a chunk cannot begin until the entire chunk has been downloaded. By using an ABR algorithm, the client can switch between different bitrates at chunk boundaries to adapt to the changing network connections.

B. Related Work

Existing video streaming algorithms can be broadly classified into four categories: buffer-based, throughput-based, control theoretic and reinforcement-learning based methods.

Buffer-based approaches [4], [9] solely consider the playback buffer occupancy to determine the bitrates. On the other hand, throughput-based algorithms [10], [11] focus on matching the video bitrate to the estimated network throughput. Researchers have also explored approaches that consider both buffer and throughput information. A straightforward approach is to combine buffer-based and throughput-based algorithms, *e.g.,* DYNAMIC [12]. A more advanced approach is control-theoretic schemes that aim to maximize QoE over a receding horizon, given the buffer occupancy, predictions of future throughput, and upcoming chunk sizes. A notable example is MPC [2]. Since MPC is sensitive to throughput estimation accuracy, more advanced methods have been proposed to employ machine learning and deep learning for throughput prediction [13] and transmission time prediction [8].

Different from control theoretic-based approaches that are prone to errors in modeling complex and stochastic networking environments, an alternative approach is to apply reinforcement learning (RL) to infer ABR algorithms by maximize QoE over an entire trajectory in a model-free way [3], [14]. To adapt RL-based ABR algorithms to heterogeneous network environments, Huang *et al.* [15] introduced the use of Meta-Reinforcement Learning to tailor ABR policy.

In contrast, our work employs constrained RL to customize ABR algorithms over mmWave 5G by maximizing bitrate while limiting performance degradation via minimizing stalls. Rather than optimizing a pre-defined weighted multiple-objective QoE/reward metric [2], [3], [8], [14], COREL enables efficient exploration of optimal reward weights, resulting in a better balance between high bitrate and low stall time.

III. Problem Setting

In this section, we highlight the challenges in streaming videos over mmWave 5G networks and outline the rationale for our proposed ABR algorithm over mmWave 5G networks.

A. Video Streaming over mmWave 5G Networks

a) ***Benefits mmWave 5G brings to video streaming:*** Recent measurement studies [6] demonstrate that commercial mmWave 5G can indeed offer ultra-high bandwidth, making it well-suited for supporting bandwidth-intensive video streaming applications, such as ultra-HD (UHD) 4K/8K videos [16]. For example, streaming 8K videos requires a bandwidth ranging from 80 to 300 Mbps. This bandwidth requirement can be easily met by mmWave 5G. Fig. 3 shows the throughput distributions of our mmWave 5G traces (see §V-A1 for details). The mean throughput of mmWave 5G is about 439 Mbps, which far exceeds the bandwidth requirements for UHD 8K video streaming. Consequently, one may expect a smooth QoE when watching UHD videos over mmWave 5G. However, the QoE of video streaming over mmWave 5G is found to fall short of the expectation [5].

b) ***Challenges mmWave 5G poses to video streaming:*** The wild fluctuations in mmWave 5G throughput, coupled with the presence of 5G "dead zones", pose two significant challenges for video streaming applications. Due to its considerably shorter wavelength, mmWave is widely recognized for its susceptibility to factors such as mobility and blockage. The throughput of mmWave 5G is highly variable over time, with fluctuations ranging from 100 Mbps to 1 or 2 Gbps due to slight changes in orientations and locations, or blockages caused by moving objects in the surroundings [7], [16].

Moreover, mmWave 5G throughput may plummet to nearly zero, leading to what is commonly known as “5G dead zones”.

The impact of mmWave 5G on the performance of ABR algorithms has been investigated in [5]. The research reveals that state-of-the-art ABR algorithms that work well under 4G do not maintain high performance under 5G. Despite the capability of existing ABR algorithms to achieve high bitrates in 5G networks, they tend to experience significantly higher stall times. *Therefore, the primary concern for video streaming over 5G lies in the video stall time.* For example, Pensieve [3], an ABR algorithm based on Reinforcement Learning (RL), outperforms all other ABR algorithms in 3G and 4G, but exhibits the highest video stall time under 5G. The poor performance of Pensieve in terms of stall time is also evident in our experimental results, as detailed in §V.

B. Constrained RL-based ABR Design over mmWave 5G

Two factors contribute to the ineffectiveness of existing RL-based ABR algorithms under 5G. First, due to the high variability in network throughput, it is difficult for RL to tell whether the observed QoE feedback of two ABR decisions differs due to the disparate network conditions, or due to the quality of the learned policy [17]. The second factor lies in the multiple-objective reward function. Typically, ABR algorithms necessitate the co-optimization of multiple objectives, such as maximizing bitrate while minimizing stalls. However, RL algorithms require a single reward value for training. Existing RL-based ABR algorithms merge these multiple objectives by utilizing specific pre-defined weighted sums, leading to their performance being inherently reliant on the chosen reward weights. In practice, this predefined trade-off (*i.e.,* the reward weights) between different sub-goals does not consistently perform well in dynamic networks, such as Facebook’s video streaming platform [18] and mmWave 5G networks [5].

To improve the performance of RL-based ABR algorithms under 5G, we propose formulating the ABR algorithm optimization as a constrained optimization task. Instead of optimizing a pre-defined weighted multiple-objective reward metric, we aim to optimize one objective (*i.e.,* maximizing bitrate) while ensuring a bounded degradation in performance across the other objectives (*i.e.,* minimizing stalls to satisfy a constraint). This approach allows us to efficiently search optimal reward weights for 5G networks, as well as to help the RL to evaluate the quality of the learned ABR policy in terms of constraint satisfaction. Further, we specifically estimate the stall-time duration in order to guide us to a policy that is more likely to satisfy the constraint unlike the existing algorithms.

IV. Design & Implementation

We first mathematically formulate ABR video streaming as a constrained optimization problem. We then describe our algorithm, named COREL, to solve this optimization problem.

A. Problem Formulation of ABR Streaming

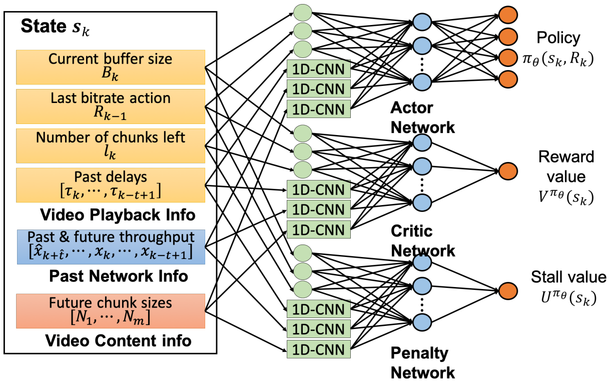

State s_k
Current buffer size B_k
Last bitrate action R_{k-1}
Number of chunks left l_k
Past delays $[\tau_k, \cdots, \tau_{k-t+1}]$
Video Playback Info
Past & future throughput $[\hat{x}_{k+\hat{t}}, \cdots, x_k, \cdots, x_{k-t+1}]$
Past Network Info
Future chunk sizes $[N_1, \cdots, N_m]$
Video Content info
1D-CNN
Actor Network
Critic Network
Penalty Network
Policy $\pi_\theta(s_k, R_k)$
Reward value $V^{\pi_\theta}(s_k)$
Stall value $U^{\pi_\theta}(s_k)$

Fig. 2: Our NN architecture overview.

The video duration is denoted as T seconds and is equally divided into K chunks. The set of available bitrates is $\mathcal{R} = \{R_1, \ldots, R_m\}$. After the $(k-1)$-th chunk is downloaded, the ABR algorithm selects the bitrate R_k from $\mathcal{R}$ for downloading the k-th chunk. Assuming the ABR starts fetching the k-th chunk at time t_k and the throughput at time t be $C(t)$, then the download time for the k-th chunk at bitrate R_k is

$$\bar{t}_k(R_k) = \inf\{\tau - t_k | \int_{t_k}^{\tau} C(t)dt \geq d(R_k)\}. \tag{1}$$

where $d(R_k)$ is the chunk size at the bitrate R_k.

The downloaded chunks are stored in a playback buffer. Let the buffer size at time t be $B(t)$. Let $B_k = B(t_k)$ be the buffer size when the player starts downloading the k-th chunk:

$$B_{k+1} = \max\{B_k - \bar{t}_k(R_k), 0\} + T/K. \tag{2}$$

The buffer has a maximum capacity B_{max}. $B(t) \leq B_{max}$, otherwise, any excess video is discarded. When $\bar{t}_k(R_k) > B_k$, there will be a stall. Stall-time for the k-th chunk is

$$U(R_k) = \max\{\bar{t}_k(R_k) - B_k, 0\}. \tag{3}$$

Constrained MPC Formulation: The optimal bitrate selection problem can be cast as a constrained model-predictive control (MPC) as the following

$$\begin{aligned} \mathcal{P}: \max_{R_k \in \mathcal{R}_k} & \sum_{k=1}^{K} E[q(R_k) - \lambda|q(R_k) - q(R_{k-1})|] \\ \text{subject to } & E\sum_{k=1}^{K}[U(R_k)] \leq \delta \end{aligned} \tag{4}$$

where $q(\cdot)$ is a video quality function of bitrate and $\lambda = 1$ to penalize the bitrate difference between two consecutive chunks. The constraint indicates that the total expected stall time should be less than or equal to a threshold δ.

B. Constrained RL-based ABR Algorithm

The constrained MPC problem is challenging to solve. Due to the unknown network throughput and hard-to-balance conflicting QoE objectives, we use the constrained RL technique to solve it, and name our ABR algorithm as COREL.

COREL utilizes a neural network (NN), shown in Fig. 2, to parametrize the policy as $\pi_\theta(R_k|s_k) \in [0, 1]$, which represents the probability that bitrate $R_k \in \mathcal{R}$ is chosen at state $s_k \in \mathcal{S}$ for k-th chunk. The input state s_k consists of three types of information. 1) Playback Information: current buffer size B_k, last selected bitrate R_{k-1}, past t chunks' download time $\overrightarrow{\tau_k}$, and the number of chunks remaining l_k. 2) Network Information: past t chunks' throughput measurements and future $\hat{t}$ throughput predictions $\overrightarrow{C_k}$ made by Lumos5G's seq2seq predictor [7]. 3) Content Information: chunk sizes $\overrightarrow{N_{k+1}}$ for each bitrate of the next $k+1$-th chunk. The output of COREL is an m-dimensional vector that represents the probabilities of selecting different bitrates at the current state, s_k.

The reward at the k-th chunk is $r(s_k, R_k) = q(R_k) - \lambda|q(R_k) - q(R_{k-1})|$. The corresponding value function is

$$V^{\pi_\theta}(s) = E\left[\sum_{k=1}^{K} \gamma^{k-1} r(s_k, R_k)|s_1 = s\right]. \tag{5}$$

where s is the initial state, γ is the discount-factor. The value function represents the cumulative reward following the policy. We further denote the realized stall-time at the k-th chunk by $u(s_k, R_k) = U(R_k)$ and the corresponding value function is

$$U^{\pi_\theta}(s) = E\left[\sum_{k=1}^{K} \gamma^{k-1} u(s_k, R_k)|s_1 = s\right]. \tag{6}$$

We seek to obtain the policy π_θ which solves the following

$$\max_{\pi_\theta} E_{s\sim\rho(\cdot)} V^{\pi_\theta}(s) \quad \text{subject to } E_{s\sim\rho(\cdot)} U^{\pi_\theta}(s) \le \delta \tag{7}$$

where $\rho(\cdot)$ is the initial distribution of state.

To solve the above CMDP problem, we consider a primal-dual-based RL approach. First, we describe the Lagrangian

$$L(\theta, \mu) = E\left[V^{\pi_\theta}(s)\right] + \mu E\left[\delta - U^{\pi_\theta}(s)\right]. \tag{8}$$

where μ is the dual variable. Interchangeably, we also denote μ as the rebuffer penalty. We then solve the min-max problem

$$\min_{\mu\ge 0} \max_{\theta} L(\theta, \mu). \tag{9}$$

C. Training Methodology

We utilize the nested loop architecture (*i.e.,* primal-dual RL) to solve the min-max problem, as it has been proved to converge under some regularity conditions [19].

Inner-loop: For a given μ_l at the l-th outer-loop, we find optimal policy for dual variable μ_l, $\pi^*_\theta(\mu_l)$, i.e.,

$$\pi_\theta = \arg\max E[V^{\pi_\theta}(s) - \mu_l U^{\pi_\theta}(s)] \tag{10}$$

The expression inside the expectation can be represented as an unconstrained composite value function $V^{\pi_\theta}_{\mu_l,U}(\cdot) = V^{\pi_\theta}(\cdot) - \mu_l U^{\pi_\theta}(\cdot)$. corresponding to the composite reward:

$$r_{\mu_l}(s_k, R_k) = r(s_k, R_k) - \mu_l u(s_k, R_k) \tag{11}$$

Hence, we can use the policy gradient mechanism to find the optimal policy for a given μ_l. In particular, we use an actor-critic-based algorithm. At step j inside the outer-loop l, the policy parameter is updated as follows:

$$\theta_{j+1} = \theta_j - \eta_\theta \nabla_\theta E[V^{\pi_\theta}(s) - \mu_l U^{\pi_\theta}(s)] \tag{12}$$

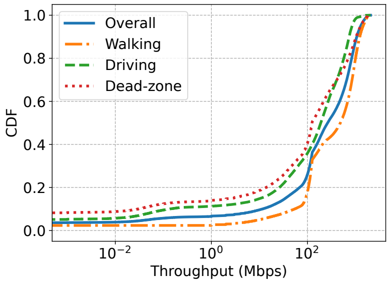


Fig. 3: The CDFs of our mmWave 5G throughput.

η_θ is the learning rate for updating θ. We use the critic network to estimate the composite value function $V^{\pi_\theta}_{\mu,U}$ and follow the standard temporal difference method to train its parameter.

Outer-loop: When we obtain θ^*, μ_l is updated as follows

$$\mu_{l+1} = \mu_l - \eta_\mu(\delta - E(U^{\pi^*_\theta(\mu_l)}(s))) \tag{13}$$

η_μ is the learning rate to update μ. We use an additional critic network to estimate $U^{\pi^*_\theta}(\cdot)$, the value function for stall time.

V. Evaluation

A. Evaluation Methodology

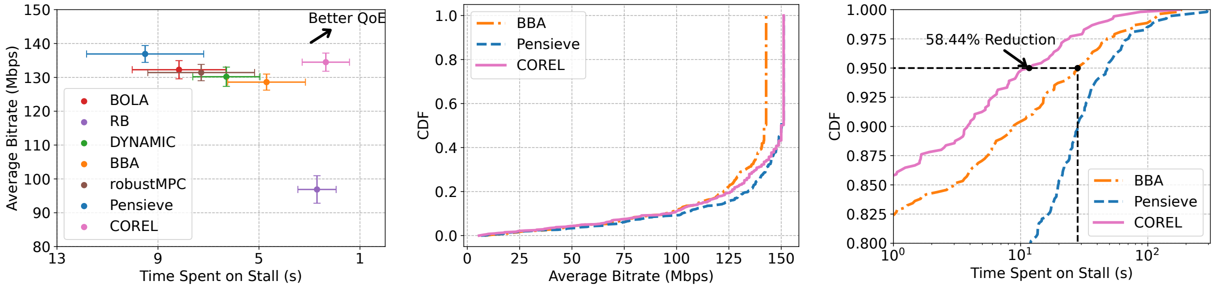
*1) **Network traces:*** The corpus of mmWave 5G traces used in this paper comprises two components: 1) the publicly available Lumos5G dataset [7] and 2) 5G traces collected by the authors. After filtering out traces where selecting the maximum bitrate is always the optimal solution or where the network is unable to support any available bitrate for an extended period, the corpus of network traces consists of 189 hours of mmWave 5G traces. The traces can be categorized into three different scenarios: 1) walking scenario, where traces were collected while walking; 2) driving scenario, where traces were collected while driving; and 3) deadzone scenario, which includes traces where the 5G throughput remains at 0 for an extended period (indicating entry into 5G cellular dead zones). Fig. 3 show the distributions of our 5G network traces. Unless otherwise noted, we use a random 80% of the traces to train both Pensieve and COREL and the remaining 20% to evaluate all ABR algorithms.

*2) **ABR algorithms**:* We compare COREL to the following state-of-the-art ABR algorithms: 1) buffer-based: BBA [9] and BOLA [4]; 2) throughput-based: simple rate-based (RB); 3) hybrid-based: DYNAMIC [12], the default ABR algorithm in the DASH player; 4) control theoretic: robustMPC [2]; 5) reinforcement learning-based: Pensieve [3] [1].

*3) **Experimental setup**:* We employ trace-driven emulation and the testbed comprises an Apache server that hosts the video and a DASH.js video client. We use an 8K video [20] from the Internet and encode it using FFmpeg in {360p, 720p, 1080p, 2K, 4K, 8K}. We implement COREL using TensorFlow [2]. We set the state input with a history length of 8 and a future length of 4. The learning rates for the actor and critic networks are 10^{-4} and 10^{-3} respectively.

[1] We compare to Pensieve because it is both the state-of-the-art RL-based algorithm and open-source.

[2] The source code of COREL and our mmWave5G traces are available at https://github.com/COREL-ABR.

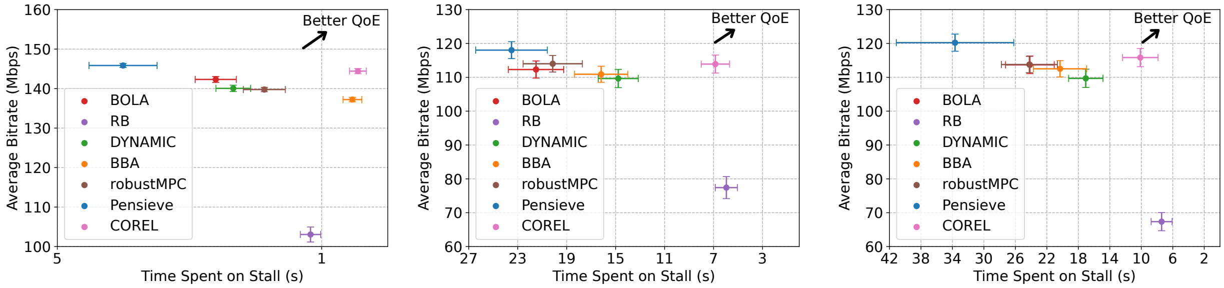


(a) Error bars of bitrate and stall time

(b) Distribution of average bitrate

(c) Distribution of stall time

Fig. 4: Error bars and CDFs of bitrate and stall time on the random test traces. Error bars show 95% confidence intervals.



(a) Walking scenario

(b) Driving scenario

(c) Deadzone scenario

Fig. 5: The bitrate and stall time tested on three different 5G network scenarios. Error bars show 95% confidence intervals.

The stall constraint δ is 0.01. The initial value of the dual variable μ is 160. To ensure a fair comparison, we use the same configurations to train Pensieve on our 5G traces.

B. COREL vs. Existing ABR Algorithms

Both Pensieve and COREL are trained using the training dataset described in §V-A. Then, all ABR algorithms are evaluated on the randomly held-out test traces. Fig. 4a shows the average bitrate and video stall time of all ABR algorithms, accompanied by error bars denoting 95% confidence intervals. COREL achieves an outstanding balance between high bitrate and low stall time, outperforming both Pensieve and BBA in this regard. Specifically, when compared to Pensieve, COREL achieves a slight decrease in average bitrate by 1.77%, but significantly reduces the time spent on stall by 75.27%. This trade-off of sacrificing a small portion of bitrate yields substantial benefits in terms of reducing stall time. When compared to BBA, COREL achieves a 4.58% increase in average bitrate while simultaneously reducing stall time by 49.96%.

We further examine Pensieve, BBA, and COREL's distributions of bitrate and stall time, as depicted in Fig. 4. The CDFs show interesting patterns. In network traces with low throughput, COREL tends to make more conservative decisions compared to BBA and Pensieve. In network traces with good throughput, COREL is more conservative than Pensieve but utilizes the available throughput more effectively than BBA. These bitrate decision patterns lead to COREL achieving significantly lower stall time compared to Pensieve and better tail performance (58.44% reduction of stall time at 95-percentile) compared to BBA.

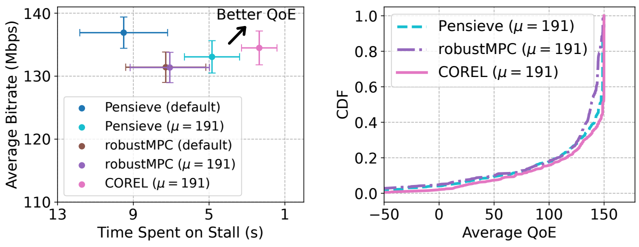
C. Generalization

In this section, we investigate the impacts of different 5G network scenarios (*i.e.,* walking, driving, and deadzone scenarios) on the performance of ABR algorithms. Note that both Pensieve and COREL are trained using randomly selected traces as mentioned earlier. Fig. 5 shows the average bitrate and video stall time of all ABR algorithms as well as the error bars representing 95% confidence intervals. COREL consistently outperforms state-of-the-art ABR algorithms by achieving an excellent balance between high bitrate and low stall time. In driving and dead-zone scenarios, compared to Pensieve, COREL achieves a slight decrease in bitrate by 3.49% and 3.66% respectively. However, it demonstrates significant improvements in stall time reduction, with reductions of 70.85% and 69.93% in the driving and dead-zone scenarios, respectively. Compare to DYNAMIC, which outperforms BBA in these two scenarios, COREL achieves a slight increase in bitrate by 3.89% and 5.60% respectively, while simultaneously reducing stall time by 53.67% and 40.63% respectively. In the walking scenario, COREL still achieves the smallest stall time than existing ABR algorithms. While both COREL and BBA exhibit stall times below 1 second, COREL selects bitrates that are 5.25% higher than BBA.

D. Impact of optimal rebuffer penalty

In this section, we analyze the impact of optimal rebuffer penalty μ on ABR algorithms that optimize for QoE metrics. To this end, we plug the rebuffer penalty value, discovered by COREL, into the QoE reward functions [3] of robustMPC and

[3] QoE $= \sum q(R_k) - \mu \sum T_k - \sum |q(R_k+1) - q(R_k)|$, where T_k is the stall time of k-th chunk at bitrate R_k and the default μ value is 160 [5].



(a) The impact on bitrate and stall time

(b) CDFs of QoE

Fig. 6: Impact of rebuffer penalty on ABR algorithms.

Pensieve, and retrain Pensieve from scratch.

Fig. 6a shows the performance of Pensieve and robustMPC with the default μ of 160 and with the μ of 191 which is attained by our approach. COREL achieves a better balance between high bitrate and low stall time compared to Pensieve and robustMPC. When $\mu = 191$ is used, COREL shows an increase in bitrate by 1.07% and 2.66% respectively, along with a decrease in stall time by 51.47% and 64.78% respectively. We observe that Pensieve is sensitive to the rebuffer penalty. The gap between Pensieve and COREL implies that depending solely on the rebuffer penalty is insufficient to effectively guide the RL algorithms to maintain low stall time and high bitrate. The reason for this could be that optimizing solely based on the feedback of QoE reward cannot distinguish whether changes in QoE are caused by the bitrate changes, stall time changes, or a combination of both. As shown in Fig. 6b, where all the algorithms use the same QoE metric, although the QoE performance of COREL and Pensieve is close, COREL outperforms Pensieve, especially in terms of stall time. This further indicates the benefits of constrained training, which provides an additional evaluation criterion by considering the satisfaction of the stall time constraint.

VI. Conclusion and Future Work

We proposed a CMDP formulation for optimizing the ABR algorithm, unlike existing approaches which consider finding a policy for a pre-defined weighted QoE reward. We propose a primal-dual RL algorithm, COREL, for simultaneously finding the policy and tuning the weight parameter corresponding to stall time. COREL outperforms state-of-the-art algorithms on various mmWave 5G network scenarios and achieves a superior balance between high bitrate and low stall time. In particular, COREL reduces the average stall time by up to 75% and reduces the tail stall time at 95-percentile by up to 58% at the similar bitrate level of the best existing schemes.

ACKNOWLEDGEMENTS

This research was in part supported by NSF under Grants CNS-1814322, CNS-1901103, CNS-2128489, CNS-2312836, CNS-2106933, CNS-2112471, CNS-1955535, Army Research Office grant W911NF-21-1-0244, Army Research Laboratory under Cooperative Agreement Number W911NF-23-2-0225, and the NJIT faculty startup grant 172884. Arnob Ghosh did this work when he was at the Ohio State University.

References

[1] Ericsson, "Mobile data traffic outlook – 5g to drive all mobile data growth," 2023. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast

[2] 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.

[3] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 197–210.

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

[5] A. Narayanan, X. Zhang, R. Zhu, A. Hassan, S. Jin, X. Zhu, X. Zhang, D. Rybkin, Z. Yang, Z. M. Mao *et al.*, "A variegated look at 5g in the wild: performance, power, and qoe implications," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 610–625.

[6] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z.-L. Zhang, "A first look at commercial 5g performance on smartphones," in *Proceedings of The Web Conference 2020*, 2020, pp. 894–905.

[7] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li *et al.*, "Lumos5g: Mapping and predicting commercial mmwave 5g throughput," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 176–193.

[8] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. A. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming." in *NSDI*, vol. 20, 2020, pp. 495–511.

[9] 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.

[10] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012, pp. 97–108.

[11] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 272–285.

[12] K. Spiteri, R. Sitaraman, and D. Sparacio, "From theory to practice: Improving bitrate adaptation in the dash reference player," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 2s, pp. 1–29, 2019.

[13] G. Lv, Q. Wu, W. Wang, Z. Li, and G. Xie, "Lumos: Towards better video streaming qoe through accurate throughput prediction," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 650–659.

[14] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, X. Yao, and L. Sun, "Comyco: Quality-aware adaptive video streaming via imitation learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 429–437.

[15] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, and L. Sun, "Learning tailored adaptive bitrate algorithms to heterogeneous network conditions: A domain-specific priors and meta-reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2485–2503, 2022.

[16] E. Ramadan, A. Narayanan, U. K. Dayalan, R. A. Fezeu, F. Qian, and Z.-L. Zhang, "Case for 5g-aware video streaming applications," in *Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases*, 2021, pp. 27–34.

[17] H. Mao, S. B. Venkatakrishnan, M. Schwarzkopf, and M. Alizadeh, "Variance reduction for reinforcement learning in input-driven environments," *arXiv preprint arXiv:1807.02264*, 2018.

[18] H. Mao, S. Chen, D. Dimmery, S. Singh, D. Blaisdell, Y. Tian, M. Alizadeh, and E. Bakshy, "Real-world video adaptation with reinforcement learning," *arXiv preprint arXiv:2008.12858*, 2020.

[19] S. Paternain, L. Chamon, M. Calvo-Fullana, and A. Ribeiro, "Constrained reinforcement learning has zero duality gap," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] A. Ferreira, "Canon eos r5 cinematic camera test in 8k raw!" 2020. [Online]. Available: https://www.youtube.com/watch?v=7JJDA134VN4