

Joint Optimization of Handoff and Video Rate in LEO Satellite Networks

Kyoungjun Park[†], Zhiyuan He^{†,1}, Cheng Luo^{†,1,2}, Yi Xu[§], Lili Qiu^{†‡}, Changhan Ge^{†,3}, Muhammad Muaz[†]

[†]The University of Texas at Austin, Austin, Texas, United States of America, {kjpark, lili, chge, mmuaz}@cs.utexas.edu

[‡]Microsoft Research Asia, Shanghai, People’s Republic of China, {zhiyuhe, luocheng, liliqu}@microsoft.com

[§]University of Science and Technology of China, Hefei, Anhui, People’s Republic of China, yi_xu@mail.ustc.edu.cn

Abstract—Low Earth Orbit (LEO) satellite communication is a promising approach to providing Internet connectivity to users in many remote areas. As videos are likely to account for most traffic in the LEO satellite network, as in the rest of the Internet, this work introduces a novel video-aware mobility management framework tailored for LEO satellite networks. Utilizing simulation models alongside real-world datasets, we show the importance of handoff strategy and throughput prediction algorithms in single-user and multi-user video streaming scenarios. Motivated by these observations, we propose a set of novel algorithms that can jointly choose the satellite and video bitrate to optimize the Quality of Experience (QoE). We first develop Model Predictive Control (MPC) and Reinforcement Learning (RL) based algorithms for a single user, and then extend them to accommodate multiple competing users that may share the same satellite. We introduce *centralized training and distributed inference* for our RL design, enabling a distributed policy informed by a global perspective. We demonstrate the effectiveness of our proposed models using trace-driven simulation and testbed experiments. We share our code and data with the research community⁴.

I. INTRODUCTION

Motivation: Recently, we have witnessed a rapid increase in Low Earth Orbit (LEO) satellite networks. Several companies, such as SpaceX, Amazon, and OneWeb, have launched or planned to launch thousands of satellites into space to build their commercial satellite networks. LEO satellite networks are attractive due to their ability to provide global network coverage at an affordable price and their lower latency and higher throughput than geostationary satellites [1], [2]. On the other hand, a LEO satellite moves at a rapid speed – it moves at around 7.5 km/s relative to ground stations, circulating the Earth every 90 to 110 minutes. This rapid movement restricts a satellite connection to a duration of only tens of seconds to three minutes, thus necessitating frequent handoffs. Zhao *et al.* [3] report that Starlink switches the primary link to another satellite approximately once every 15 seconds. Therefore, such frequent handoffs call for an effective handoff strategy to ensure smooth and high performance.

Note that although the handoff frequency in LEO satellite communication is high, the satellites’ movement follows a deterministic pattern. This characteristic is very different from traditional Wi-Fi and cellular networks, where mobility is less predictable. Such a predictable mobility pattern presents a valuable opportunity to optimize handoff strategies, yielding significant potential benefits.

Meanwhile, we also observe a rapid growth in video streaming traffic on the Internet. Video streaming has become the majority of Internet traffic. LEO satellite providers offer Internet access to remote regions (e.g., Africa, remote villages), where satellite links are their only way to access the Internet. We expect remote users to have similar traffic patterns as traditional Internet users to enjoy video streaming, which accounts for most Internet traffic.

Studies have shown that video quality is crucial to retaining viewers [4], [5]. Adaptive video streaming addresses fluctuating network throughput, where the sender divides a video into several chunks and encodes each video chunk at several different data rates [6]. Clients dynamically select the appropriate bitrate based on current network conditions. A variety of algorithms have been proposed to select the video bitrate to optimize the video Quality of Experience (QoE), which is determined by three major factors: video quality, rebuffering time, and smoothness of quality. Some use optimization while others use Reinforcement Learning (RL) to adapt the video bitrate.

Our approach: While video streaming and LEO network are important topics, little work studies them jointly. This motivates our exploration of this topic. We first conduct simulations and real-world measurements to identify key challenges and opportunities in the video streaming process in LEO networks. Our results indicate that (i) throughput pattern is important in the video QoE and we should explicitly incorporate the LEO satellites’ movement along with the environmental factors for prediction, (ii) widely used handoff algorithms, such as maximizing Received Signal Strength (RSS) or maximizing the satellite serving time, do not yield good video QoE, and (iii) multiple users’ sharing the satellite’s bandwidth further complicates the design of video rate adaptation.

To address these challenges, we observe a strong interaction between handoff and Adaptive Bitrate (ABR) algorithms and propose novel methods that jointly optimize satellite selection

Corresponding author: Lili Qiu, liliqu@cs.utexas.edu

¹ These authors contributed equally.

² Cheng Luo is currently a post-doctoral researcher at California Institute of Technology, Pasadena, CA, USA

³ Changhan Ge is currently a research engineer at AT&T Labs - Research, Bedminster, NJ, USA.

⁴ <https://github.com/kyoungjunpark/Joint-LEO>

and video bitrate. We show that by explicitly incorporating the impact of satellite selection into video QoE, we can effectively balance the tradeoff between handoff numbers and RSS. Moreover, to support multiple users competing for limited satellite throughput, we design novel algorithms to jointly select satellites and video bitrates for all users.

Our contributions can be summarized as follows:

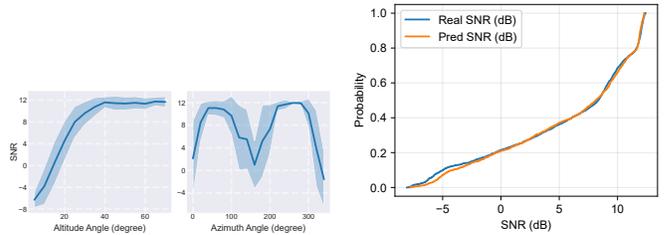
- We first use measurements to identify challenges of video streaming in LEO satellite networks. We show that it is important to design a handoff strategy considering video performance. We also show the importance of throughput prediction in both single-user and multi-user scenarios.
- We develop the first algorithms that jointly manage mobility and video bitrate in LEO networks. Our suggested algorithms are based on Model Predictive Control (MPC) and RL. In particular, we develop a *centralized training and distributed inference-based RL algorithm* to explicitly consider the interaction between multiple satellites and ground stations while supporting practical use.
- We conduct both trace-driven simulation and testbed experiments. Our results show that joint video rate and satellite selection lead to 18% to 68% QoE improvement across different settings.

II. RELATED WORKS

LEO satellite communication. LEO satellites orbiting at 250–1000 km offer lower delay than Geostationary Earth Orbit (GEO) and Medium Earth Orbit (MEO) systems due to their proximity to Earth. While electromagnetic signals propagate at light speed in space, optical fiber slows signals by approximately 31% [7]. As a result, LEO constellations can achieve round-trip latency near 20 ms [8], enabling real-time applications such as video conferencing and cloud gaming. For long-distance communication (beyond 3000 km), LEO satellites can outperform terrestrial fiber links in latency [8]. Their low altitude also supports the use of high-frequency bands (e.g., K_u , K_a , THz), which allow higher throughput.

Recent works focus on efficient networking and scheduling for LEO systems. Starfront [9], Orbitcast [10], and L2D2 [11] improve downlink performance using trajectory-based scheduling, satellite visibility estimation, and ground station coordination. SkyTube [12] jointly optimizes satellite selection and video resolution using local user states.

Bitrate adaptation. ABR algorithms dynamically select video quality to maximize QoE under varying networks. Rate-based schemes, such as FESTIVE [13] and CS2P [14], rely on predicted throughput. Buffer-based approaches, e.g., BOLA [15], use buffer occupancy to guide decisions. Hybrid methods, including RobustMPC [16], combine both metrics for better performance. RL has been applied to ABR via Pensieve [17], which uses the A3C algorithm [18] to adapt to unseen network traces. QARC [19] enhances this by learning representations of video-aware states using CNNs and RNNs. Fugu [20] replaces MPC throughput predictors with learned models. SENSEI [21] incorporates human quality sensitivity into the



(a) SNR vs. altitude/azimuth (b) Predicted vs. measured SNR

Fig. 1: SNR characteristics of NOAA satellites.

adaptation logic. Other efforts explore multipath-aware ABR (PERM [22]) and low-latency streaming (BitLat [23]).

Satellite handoff. Due to their high orbital velocity (7.5 km/s [24]), LEO satellites periodically leave the field of view of ground users, necessitating frequent handoffs. These transitions introduce 150–300 ms of disconnection at the physical layer [25], with compounding effects on transport and application layers [26]. Earlier studies addressed mobility in LEO systems from an IP- or circuit-switching perspective [27], [28]. Recent work by [29] proposes trajectory-aware handoff timing algorithms, simulating the impact of mobility and antenna tracking errors on performance.

III. BACKGROUND AND MOTIVATION

This section presents key challenges and opportunities in supporting video streaming over LEO satellite networks, motivating our algorithmic design.

A. Predictability of Satellite Signal

LEO satellites follow predictable orbits, allowing accurate signal forecasting. Although Starlink terminals lack RSS reporting, we measure the Signal-to-Noise Ratio (SNR) from NOAA satellites with similar motion and altitude.

Fig. 1a shows a strong correlation between SNR and satellite angles. We develop an ML-based SNR predictor achieving a Mean Absolute Error (MAE) of 0.84 dB (Fig. 1b). This predictability enables ABR algorithms [13], [14], [16] to map SNR to data rates and expected video bitrate, enhancing streaming performance.

B. Video Performance in LEO Network

We study video QoE under real Starlink usage in two environments: one free of obstructions and another with visible obstructions (e.g., buildings or trees). As shown in Fig. 2, obstructions severely degrade QoE by causing throughput drops below 0.1 Mbps.

We use popular ABR algorithms such as RobustMPC [16] and Pensieve [17], and apply the following QoE metric [16]:

$$\text{QoE} = \sum_{k=1}^N [\mu_1 Q(R_k) - \mu_2 T(R_k)] - \mu_3 \sum_{k=1}^{N-1} |Q(R_{k+1}) - Q(R_k)| \quad (1)$$

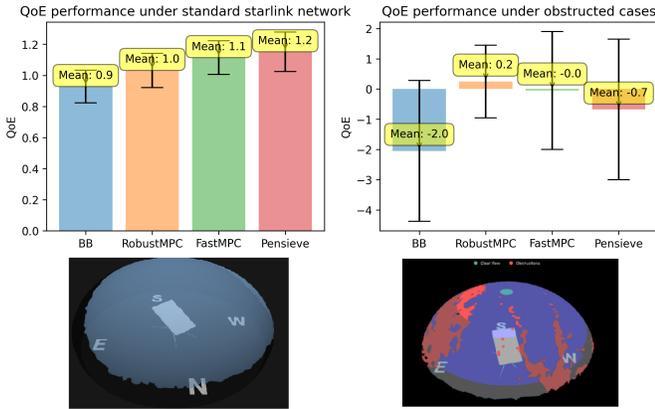


Fig. 2: Video QoE under unobstructed (left) and obstructed (right) settings, with visibility maps from the Starlink app.

where R_k is the bitrate for chunk k , $Q(\cdot)$ is the bitrate-to-quality mapping, $T(\cdot)$ is the rebuffering time, and μ_i are weighting parameters, respectively.

By utilizing SNR predictions, ABR algorithms can proactively reduce the video bitrate prior to signal degradation, allowing buffer accumulation and enhancing robustness. Moreover, jointly switching to an alternative satellite before disconnection helps sustain consistent throughput and QoE.

C. Satellite Handoff Selection

Due to rapid satellite motion (7.5 km/s [24]), frequent handoffs are inevitable. Each handoff incurs 150–300 ms delay [25], amplified at higher layers [26], sometimes leading to 5-second buffering [3]. Starlink currently does not expose handoff control to users.

Common handoff strategies include selecting satellites with either maximum visible time or maximum RSS. The former reduces handoff frequency but may sacrifice signal quality; the latter improves signal quality but increases handoff overhead. Hence, we integrate handoff decisions into the QoE model and optimize satellite selection jointly with bitrate decisions.

D. Throughput Sharing in LEO Network

Throughput sharing among users significantly impacts QoE. Fig. 3 shows measurements under three settings using Starlink: a single dish achieves 120 Mbps; two co-located dishes sharing the same satellite reduce throughput to 50 Mbps each; but directing the dishes to connect to different satellites improves aggregate throughput.

These results highlight the need for coordinated satellite selection in multi-user environments. We explore both distributed and centralized strategies for joint satellite and bitrate selection to optimize network-wide video QoE.

IV. OUR APPROACH

We formulate the joint optimization of video bitrate adaptation and satellite selection in §IV-A. Then, we describe our joint optimization for a single user in §IV-B and joint optimization for multiple users in §IV-C.

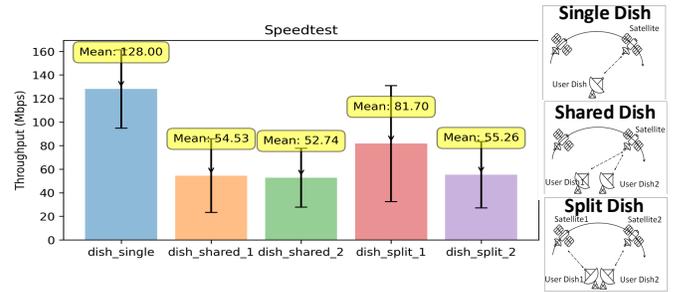


Fig. 3: Throughput with (i) one dish, (ii) two dishes in the same direction, (iii) two dishes in different directions.

A. Joint Optimization Problem Formulation

To support video streaming in LEO networks, one option is to adopt the same strategy as on the Internet, which decouples the video bitrate selection and satellite selection. While simple, this strategy leads to sub-optimal performance since satellite selection significantly impacts video streaming QoE. Therefore, we propose having clients perform joint satellite and video bitrate selection.

Moving the handoff decision from the satellite to the end host is consistent with the trend in cellular networks, which have also moved from network-based handoff to client-based handoff, to not only reduce latency but also offload base stations. A LEO client can determine when to handoff and which satellite to handoff to because the satellites’ trajectories are known in advance according to [30]. Moreover, the handoff can be performed efficiently on a per-packet basis by applying the corresponding beamforming weight to the received signals across its antenna array and passing the combined signal to the decoder for further processing. As the client knows its performance objective, having it select the satellite and video rate improves performance.

We define the QoE metric as follows to capture the impact of the satellite selection:

$$\text{QoE}_{\text{SAT}} = \sum_{k=1}^N [\mu_1 Q(R_k) - \mu_2 T(R_k, \text{Sat}_k)] - \mu_3 \sum_{k=1}^{N-1} |Q(R_{k+1}) - Q(R_k)| \quad (2)$$

Our joint ABR algorithm tries to optimize the QoE in Eq. 2. The only difference from existing separate ABR algorithms is that Sat_k is considered an optimization variable.

B. Single-user QoE Optimization Algorithms

In this section, we present two methods, MPC and RL, in LEO networks and analyze their respective characteristics.

1) *Joint MPC Algorithm:* MPC is effective for video bitrate adaptation on the Internet [16], [31]. It uses predicted throughput to optimize the QoE for the next few video chunks. It finds the best QoE by calculating all possible combinations of video quality for several future video chunks. Therefore, its performance highly depends on the quality of throughput prediction. To accommodate throughput prediction errors, there are several variants of MPC. RobustMPC improves the

Algorithm 1 Joint pruned MPC

Require: Sat : satellite, t : timestamp, k : video chunk index, C, \hat{C} : throughput, Q : QoE reward, R : bitrate, B : buffer size, Sat_c : the current satellite, H : handoff point

- 1: $\hat{C}_{Sat_c, t_k} = ThroughputPred(C_{Sat_c, [t_{k-N}, t_k]})$
- 2: $Q_{best}, R_{best} = f_{mpc}(R_{k-1}, B_{k-1}, \hat{C}_{Sat_c, t_k})$
- 3: **for** Sat_n in $VisibleSatellites(t_k)$ **do**
- 4: $\hat{C}_{Sat_n, t_k} = ThroughputPred(C_{Sat_n, [t_{k-N}, t_k]})$
- 5: **for** $H = 0$ to $F - 1$ **do** $\triangleright F$: future prediction length
- 6: $Q, R = f_{mpc}^{sat}(R_{k-1}, B_{k-1}, H, \hat{C}_{Sat_c, t_k}, \hat{C}_{Sat_n, t_k})$
- 7: **if** $Q_{best} < Q$ **then**
- 8: $Q_{best} = Q$
- 9: $R_{best} = R$
- 10: $Sat_{best} = Sat_n$
- 11: $H_{best} = H$
- 12: **end if**
- 13: **end for**
- 14: **end for**
- 15: **return** $Q_{best}, R_{best}, Sat_{best}, H_{best}$

robustness of prediction against error by estimating prediction errors and using $C = \frac{Predict}{1 + \max\{error\}}$ as input. We adopt RobustMPC in our problem due to its resilience against errors.

We propose two satellite-specific MPCs: joint exhaustive MPC and joint pruned MPC. Joint exhaustive MPC considers all combinations of satellites and bitrates for the future chunks download, which is the same logic as traditional MPC but additionally includes the satellites in the combination. However, this approach is computationally heavy due to its significant resource requirements. Its time complexity for a single calculation is $O(|R|^F \times |Sat|^F)$ where $|R|$ is the number of data rates, $|Sat|$ is the number of satellites, and F is the number of future chunks considered. This cost is too high to run in real time. Therefore, we propose a joint pruned MPC assuming only one handoff during the next F video chunks in the optimization horizon. This is a reasonable assumption since F is generally small (e.g., $F = 5$ in existing work) and multiple handoffs in such a short time window are expensive. Joint pruned MPC significantly reduces computational overhead over joint exhaustive MPC. Its time complexity is $O(|R|^F \times |Sat| \times F)$ where F represents the possible handoff point. We use joint pruned MPC in evaluation.

Algorithm 1 shows a joint pruned MPC that considers both bitrates and satellites. The video player starts downloading a video chunk k at t_k . We then predict the future throughput of the connected satellite S_c using $ThroughputPred()$. We employ two prediction methods and compare their performances: (i) Harmonic mean over historical throughput; (ii) A model similar to [11] which takes elevation angle, azimuth angle, and weather information as input, and leverages machine learning models to predict future throughput. With the predicted throughput, $f_{mpc}()$ applies the MPC algorithm. This algorithm tries to find the best QoE by iterating over all satellite candidates $VisibleSatellites()$ and handoff points H when downloading future chunks. At each handoff point H , we use $f_{mpc}^{sat}()$ to compute the utility by taking into account

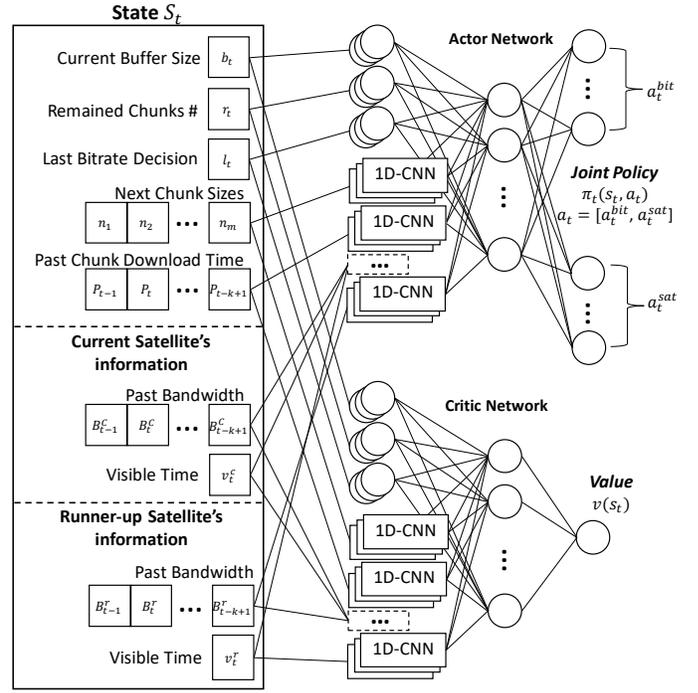


Fig. 4: PPO-based RL generates joint optimization policies.

the handoff delay and throughput change when switching from Sat_c to Sat_n at t_{k+H} .

2) *RL-based Algorithm:* Our optimization problem is well-suited to an RL framework, where an agent learns to select the satellite and video bitrate that maximizes QoE. We use proximal policy optimization (PPO) [32], an actor-critic method that trains a critic to estimate the value function and an actor to optimize the policy while constraining policy updates for stability. Fig. 4 illustrates the network with state space S , value function v , action space a , and state transition probability π .

State: At time t , state $S_t = (b_t, \gamma_t, l_t, n_m, P_t, B_t, v_t)$ includes current buffer level b_t , remaining chunks γ_t , last chunk bitrate l_t , available next chunk sizes n_m , download times P_t , throughput measurements B_t for two satellites, and visible time v_t . We extend prior work [17] by adding B_t and v_t for both current and runner-up satellites, essential for joint bitrate and handoff decisions. Only past states are input to predict actions implicitly without separate predictors.

Action: Action $a_t = [a_t^{bit}, a_t^{sat}]$ selects bitrate and satellite.

Reward: The video QoE (Eq.2) is the reward to optimize.

Policy: Policy $\pi_t(s_t, a_t)$ gives probabilities of choosing bitrate and satellite from visible candidates. For efficiency, we limit candidates to two satellites (best and runner-up) in evaluation.

Policy update: The RL agent learns to maximize QoE by receiving rewards r per chunk. The critic estimates gradients guiding actor updates via the PPO policy gradient.

Training process: The algorithm collects user states and satellite throughput data, calculates individual QoE, tunes hyperparameters (learning rates, weights), recalculates QoE to avoid degrading other users, makes decisions based on updated QoE, and repeats this cycle until the session ends or user state changes significantly.

C. Multi-user QoE Optimization Algorithms

In §IV-B, we finally introduce several optimization algorithms for a single-user scenario. In practice, multiple user terminals can share the same satellite and compete for limited network resources. As shown in §III-D, the throughput of each user significantly decreases when sharing with others.

There are several ways to address this issue: (i) Let users independently make their own decisions using the single-user algorithm described in §IV-B. The throughput is affected by cross traffic, making this approach sub-optimal and potentially unstable, as users optimize their own utility rather than the global one. (ii) Use a centralized algorithm that considers information from all users and makes decisions that maximize global utility; however, it faces deployment issues in practice, as it is challenging to control all users' selections of satellites and video rates. Motivated by the limitation of the above options, (iii) we design the RL that uses other users' states as input to optimize the global objective but only takes an action for a specific user at a time. This improves global coordination while remaining feasible to deploy, since it avoids controlling all users simultaneously. Last but not least, this RL design involves collecting other users' information in real time, which may still be difficult in some deployments. Therefore, (iv) we introduce *centralized training and distributed inference* RL design. This approach uses all users' states as the input during training but only uses the current user's state during inference, which not only simplifies the deployment but also allows us to explicitly consider the interaction between multiple satellites and ground stations.

Our multi-user algorithms primarily aim to optimize the QoE for users engaged in video streaming. However, in real-world scenarios, LEO networks support a diverse range of applications beyond video streaming, including online gaming and web browsing. To better reflect these practical environments, we incorporate multiple users with dynamic resource allocation mechanisms, simulating the diverse network demands of different applications. This approach ensures a more realistic and adaptable optimization framework.

1) *Multi-user RL (Joint RL (L))*: We apply the single-user PPO-based RL to the multi-user scenario. Our multi-user process can be represented by the following four tuples: (S_t^j, A^j, π^j, R^j) , where j is the user id, S_t^j represents the user j 's state, A is the distributed action space, π is the state transition probability, and R is the local reward value. The decision-making process of a single-user RL can be directly applied to multi-user optimization. This RL uses only the current user's state, so we call it the local state. It uses the throughput history of the current user's satellite to implicitly learn the network condition when sharing its satellite with other users.

For multi-user training and testing, we use a critic network for distributed training. The input state of the critic network is a distributed environment (*i.e.*, S_t^j for each user). The critic network estimates the value of action decisions generated by the actor network. The state transition probability π is then used to update the subsequent actor steps, evaluate each

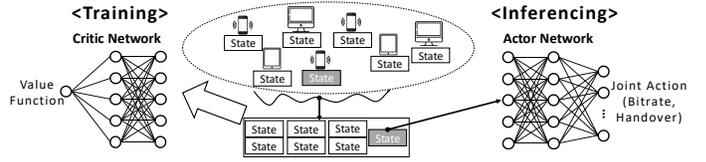


Fig. 5: PPO-based RL algorithm generates joint optimization policies with centralized training and distributed inference.

actor, and guide their parameter updates. The parameters and gradients of both networks are shared by all agents.

2) *Multi-user RL with Global Status (Joint RL (G))*: This approach leverages all users' states to maximize all users' overall QoE, hence called Joint RL with global information (Joint RL (G)). This RL algorithm works by substituting the input states in the multi-user RL design with the following: (S_t, A^j, π^j, R^j) for each user j . The only difference is that S_t represents all users' states, not just one user. We feed all users' information to the critic network to train a single global policy. During training, the centralized critic network is used to update the actor networks of all users. The gradient of the critic network is computed based on the actions taken by all users, and updates in the actor network are based on gradients in the critic network. Since it requires global information not only for training but also for the inference phase, it requires users to frequently exchange their states with other users who may share the same satellites.

3) *Multi-user RL with Centralized Training and Distributed Inference (Joint RL (L+))*: Inspired by the asymmetric actor-critic structure from the RL community [33]–[35], we use a centralized critic architecture and a decentralized actor architecture, as shown in Fig. 5. During training, the centralized critic network takes the global state as input and outputs a single value for each state, which is used to update the actor networks of all users. This speeds up convergence and improves performance. During inference, the decentralized actor network takes the local state of each user as input and outputs a local action for that user. In other words, *centralized training and distributed inference* have a hybrid concept of considering the global objective during training and making decisions at the user-level state during inference.

4) *Centralized MPC*: So far, we have described several RL algorithms that can run in practice. We also consider the centralized MPC algorithm as another baseline. This algorithm assumes we have full control of all users' actions. While this assumption is strong and may not hold in practice, this is an interesting baseline. Without prediction error and with a large enough optimization horizon, the oracle centralized MPC establishes the upper bound for multi-user video QoE. We extend the single-user MPC to a centralized MPC by (i) treating all users' selected satellites and video bitrates as the optimization variables and (ii) taking the sum of all users' utilities as the objective. Since MPC calculates all the possible combinations, the search space increases exponentially with the number of users. To reduce running time, we prune the satellites with low throughput.

V. SYSTEM IMPLEMENTATION

To deploy our proposed approach in real-world scenarios, we address a key practical challenge: the absence of a native interface in Starlink that allows clients to actively select or switch between satellites. To overcome this limitation, we develop a custom client-server architecture that enables both bitrate selection and satellite handoff at the client side.

We emulate the satellite network using TCP sockets to represent distinct links between users and satellites. Upon a client’s connection to a satellite, a dedicated TCP socket is created on the server side and managed by a satellite controller. This controller enforces bandwidth constraints by rate-limiting the socket according to throughput traces collected from real-world Starlink deployments or from synthetic traces generated for experimentation. Note that the Starlink traces represent end-to-end throughput, encompassing both uplink and downlink, as they are collected from the connection between the video server and end users. When replayed in our system, they accurately emulate real-world conditions.

When a client initiates a satellite handoff, the system closes the current socket and establishes a new one under a different satellite controller. Multiple clients may share a satellite, in which case their throughput is fairly allocated by the shared controller. To support multi-agent reinforcement learning (Joint RL(G)), clients report their internal states—such as buffer occupancy and download speed—to the server after each video chunk download. In response, the server returns the latest states of all other clients via a separate API call, enabling informed joint decision-making.

RL implementation. Our reinforcement learning model takes as input a sequence of past network measurements. These include download times and throughput values for the last 8 chunks, which are processed through densely connected neural network layers with 128 neurons and 1D convolutional layers with 128 filters (kernel size 1, stride 1). These design choices are guided by a microbenchmark study.

The actor network uses a softmax activation at the output to generate a probabilistic policy over actions, while the critic network outputs a scalar value via a linear layer to estimate state values. Both networks use a learning rate of 10^{-4} , and we set the discount factor γ to 0.99 to balance short-term and long-term performance. To control the exploration-exploitation trade-off, we incorporate an adaptive entropy weight that decays with training progression.

Multi-user scenario. To evaluate the system under realistic conditions, we simulate a network environment with 20 concurrent users performing video streaming or other background tasks. For users not engaged in video streaming, we introduce dynamic resource allocation to emulate heterogeneous and fluctuating network usage patterns.

To scale our reinforcement learning framework to multi-user settings, we adopt an encoder-decoder architecture within the neural network. This design allows the model to capture inter-user interactions efficiently and produce joint action decisions. While our current implementation targets scenarios with 20

users, the framework is extensible to larger populations by proportionally increasing model capacity.

VI. EVALUATION

We evaluate our proposed MPC and RL algorithms through extensive experiments in both single-user and multi-user scenarios, using simulation and testbed setups described in §V. The video used is in 1080p, segmented into 49 chunks of 2 seconds each, and encoded at three bitrates: 300, 1200, and 2850 kbps. While real-world video servers typically offer six or more bitrates, we use three levels to manage the complexity of centralized MPC. To demonstrate scalability, we additionally evaluate **Joint RL (L+)** with six bitrates: 300, 750, 1200, 1850, 2850, and 4300 kbps. In the QoE function, the weights μ_1 , μ_2 , and μ_3 are set to 1, 4.3, and 1, respectively. The link RTT is 80 ms, and the handoff delay is 200 ms.

A. Satellite Network Datasets

We create three datasets: simulated, NOAA, and Starlink, each split into training and test sets. The simulated dataset models Starlink phase I satellite trajectories over 24 major cities worldwide using a free space path loss model to simulate throughput dynamics. The NOAA dataset contains measured SNR data from NOAA satellites, while the Starlink dataset comprises end-to-end measured throughput data from a real Starlink link between a client and a video server.

For MPC evaluation, algorithms are directly applied to test sets. For RL methods, models are first trained on the simulated dataset’s training set, then fine-tuned on the training set of NOAA or Starlink prior to testing.

The simulated throughput b_t is generated as $b_t = \alpha b_{\max} \frac{d_{\min}^2}{d_t^2} + \epsilon$, where b_{\max} is the maximum throughput of a Starlink satellite (Mbps), d_{\min} is the satellite altitude, d_t is the distance to the client, α is a scaling factor to simulate resource limits, and $\epsilon \sim \mathcal{N}(-2, 1)$ is Gaussian noise. We generate 24 traces per city lasting 10 minutes each; seven cities form the test set, and the rest are used for training.

Due to difficulty obtaining SNR from Starlink devices, we use NOAA satellites (NOAA-15, NOAA-18, NOAA-19) for SNR data. These LEO satellites have altitudes (808-854 km) and velocities (7.4 km/s) comparable to Starlink’s (550 km altitude, 7.6 km/s velocity), making NOAA traces suitable for handoff evaluation. We collect SNR and throughput traces from major US and Chinese cities, mapping elevation angle to SNR. Each Starlink satellite pass is assigned an SNR from a randomly chosen NOAA trace using this mapping, from which throughput is derived. The NOAA dataset contains 24 training and 12 test traces with no overlap.

The Starlink dataset is collected via iperf downloads from a Starlink RV terminal to a video cloud server in the US, capturing uplink, downlink, and terrestrial links. Logs are sampled every second over 13 hours. To simulate multiple satellites, we apply time-division multiplexing by converting a 16-minute log into four 4-minute logs, generating 48 multiplexed traces from the 13-hour log. Forty logs form the training set, and eight form the test set.

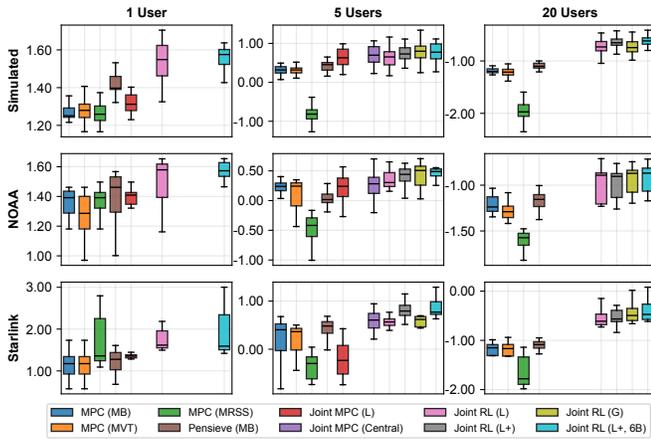


Fig. 6: QoE results of our models (i.e., MPC, RL). The results are from simulated, NOAA, and Starlink datasets. All models use three bitrate levels for fair comparison, except Joint RL (L+, 6B). The number of users refers to how many users are streaming video out of 20 users.

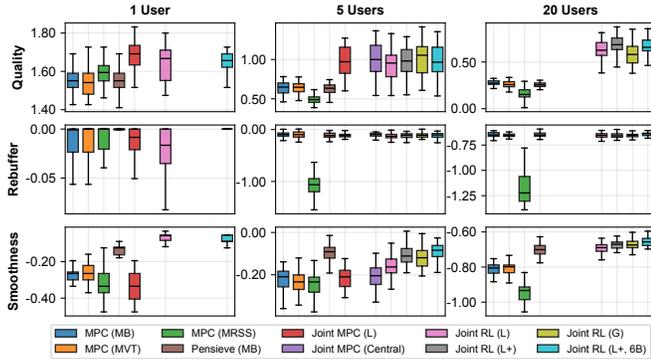


Fig. 7: QoE breakdown with (1) quality reward, (2) rebuffer penalty, and (3) smoothness penalty. The results are from the simulated dataset. All models use 3 bitrate levels for fair comparison except Joint RL (L+, 6B). The number of users refers to how many users are streaming video out of 20 users.

B. Baseline Methods

Our methods can select satellites and bitrates simultaneously. We compare them with popular satellite selection strategies and ABR algorithms. The following satellite handoff strategies are tested:

- Maximum Visible Time (MVT): Switch to the satellite with the longest visible time when the currently connected satellite is about to leave the client’s horizon.
- Maximum Received Signal Strength (MRSS): Switch to the satellite with the highest signal strength.
- Maximum Bandwidth (MB): Switch to the satellite with the maximum available bandwidth when the currently connected satellite is about to leave the client’s horizon.

C. Simulation Results

RobustMPC [16] and Pensieve [17] are used in ABR algorithms. For simplicity, we refer to RobustMPC as “MPC”. The satellite handoff strategy and the ABR algorithm are combined

and tested. For example, MPC (MRSS) uses MRSS to select satellites and RobustMPC to decide bitrates. We pair Pensieve with the MB due to its superior performance. We highlight **Joint RL (L+)** in bold to indicate that it is our final proposed model. We experiment with three scenarios involving 20 users: (i) 1 user streaming video while 19 users engage in other applications, (ii) 5 users streaming video while 15 users engage in other applications, and (iii) all 20 users streaming video.

1) *Overall Comparison*: The simulation results on three different datasets with different user numbers are presented in Fig. 6. The proposed joint methods consistently outperform separate methods, particularly when multiple users are present. Compared to the best result achieved by the separate selection, joint selection methods can improve QoE by 18%, 68%, and 57% for 1 user, 5 users, and 20 users, respectively, in the Starlink dataset. The substantial growth in QoE improvement with more users stems from our network resource allocation: while a single user enjoys ample resources, significant resource contention arises when three or more users share the network. In the simulated dataset, we train in 17 cities and test in 7 other cities. Our RL methods generalize well to unseen cities. Overall, the joint RL (G) has the best performance. Joint RL (L) and **Joint RL (L+)** are also competitive but slightly worse than Joint RL (G). Note that **Joint RL (L+)** always yields a better QoE than Joint RL (L), ranging from 0.8% to 25%. It shows that our *centralized training and distributed inference approach* effectively considers both sides of the satellite network: periodicity of satellite movement and irregular network throughput.

We present the breakdown of QoE into three terms (quality, rebuffer, and smoothness) in Fig. 7. Based on Fig. 7, separate selection methods generally produce lower-quality rewards when compared to joint selection methods. This can be attributed to a mismatch between the satellite selection strategy and the video QoE. Joint selection methods can actively change to a new satellite, which potentially brings QoE improvement. MB is the best handoff strategy to maximize the throughput among separate selection schemes, but it is still worse than the joint selection in quality score.

It is observed that MPC-based methods are often less smooth than RL-based ones, possibly due to the limited optimization window of MPC. This limitation also exists in our proposed Joint MPC method. Thus, in the 1-user scenario, although Joint MPC (L) produces higher quality rewards than Pensieve (MB), its final QoE score falls behind that of Pensieve (MB) due to worse smoothness. Due to the limited calculation window in MPC, we can see that the performance of Joint MPC (Central), which knows the network status within the window, is similar to or even underperforms our Joint RL.

Pensieve (MB) rarely picks the highest bitrate, even though it has a sufficient buffer. Hence, it has lower quality scores than our proposed Joint RL, as shown in Fig. 7. We interpret Pensieve (MB) as making a conservative decision, as it does not jointly consider satellites. When Pensieve (MB) increases the video quality due to enough buffers, it encounters many cases where a rebuffering penalty occurs due to sudden drops

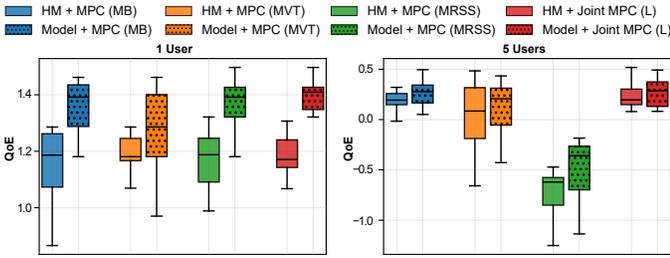


Fig. 8: QoE results of MPC methods on the NOAA dataset with and without the model prediction. "HM" indicates the harmonic mean, while "Model" uses ML for prediction.

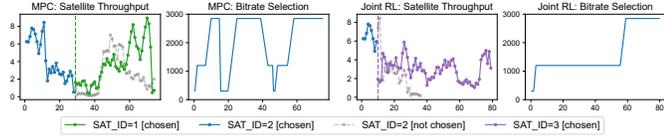


Fig. 9: An obstruction scenario, with the X-axis denoting time (sec). Initially, both Joint MPC (L) and **Joint RL (L+)** connect with satellite 2. However, satellite 2 experiences an obstruction period from 15s to 40s. Joint MPC (L) switches to a different satellite around 30s with significant bitrate fluctuation. Conversely, **Joint RL (L+)** transits to an alternative satellite when disturbed, effectively maintaining stable bitrates.

in satellite throughput.

We find that Joint RL actively decides the handoff to achieve a high QoE, *i.e.*, 2 to 3 times more handoffs than the MB strategy. Unlike Pensieve (MB), Joint RL consistently selects the highest bitrate when the buffer is sufficiently full. Interestingly, Joint RL does the handoff only when the buffer is enough, which indicates a low possibility of rebuffering. Conversely, when the buffer is sufficient, Joint RL actively performs actions to achieve high throughput, such as performing multiple handoffs within a few seconds.

By incorporating global status, Joint RL (G) achieves better results than Joint RL (L). Joint RL (G) learns interesting policies maximizing the overall QoE of all users, even if it slightly degrades one user. Joint RL (G) learns to tactically establish each user's role, such as determining handoff frequency or video quality. For example, one user does a lot of handoffs when encountering resource sharing and only maintains modest quality downloads. This allows other users to download the highest quality without resource contention. Impressively, our RL models learn these policies to optimize the overall QoE without any special a priori knowledge.

Compared to the simulation and Starlink trace for 5 users, the MPC models have long tails in the boxplot, indicating huge variances in QoE. This is because the real trace contains unstable latencies at specific timestamps, *i.e.*, handoff. To cope with this unexpected case, users changed the quality abruptly as shown in the smoothness in Fig 7. In contrast, the RL variants, including our proposed Joint RL (L+), yield highly stable QoE results, indicating that they can handle harsh network conditions more robustly. Note that a sudden one-time quality change incurs a severe degradation of QoE.

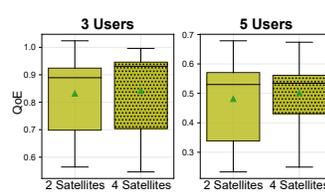


Fig. 10: **Joint RL (L+)** on NOAA dataset varying satellite selection coverage.

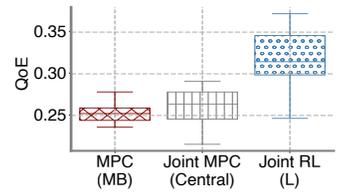


Fig. 11: QoE results of three models for the multi-session problem (3 sessions).

2) *Obstructions*: In Sec III-B, we highlight that obstructions decrease video QoE significantly in a satellite network. We find that our proposed methods can effectively handle the obstructed scenario. In the Starlink dataset, sometimes bandwidth drops abruptly, degrading QoE as shown in Fig. 2. If the bandwidth falls below the 25th percentile for over 10 seconds, it is considered an "obstruction period". In all, 158 obstruction periods were identified in the dataset, accounting for 5.8% of the total duration. Our Joint RL effectively handles these scenarios as shown in Fig. 6, and we provide a case study in Fig. 9. Unlike other methods that have bitrate fluctuations when faced with obstructions, RL methods smoothly switch to a different satellite and maintain consistently high bitrates.

3) *Effects of throughput prediction*: We employ an ML-based model similar to [11] to predict the throughput of the satellite using the NOAA dataset. The input features for the model are the azimuth angle, elevation angle, weather status, and the satellite's movement. The model uses a stacked ensemble of gradient-boosting decision trees and NN models to predict the future throughput, achieving a mean absolute error of 0.1304 *Mbps* on the test set.

Our prediction model is useful for throughput-based ABR algorithms. We compare the QoE of MPC-style methods with and without our model's predictions to evaluate its efficiency. Specifically, we compare the performance of MPC methods that use our model's predictions with those that use the harmonic mean of historical throughput to select bitrates, considering several handoff strategies.

As shown in Fig. 8, model prediction leads to improved QoE for both traditional MPC methods and our proposed joint MPC. The throughput prediction enables better bitrate selection for traditional MPC methods, while for our proposed methods, it enables not only better bitrate selection but also better satellite selection.

4) *Consideration of multiple satellites*: Our models consider 2 satellites in the decisions. Fig. 10 shows how much the QoE can be improved by considering more than 2 satellites. Since there are up to 4 visible satellites in NOAA, we evaluate the **Joint RL (L+)** by extending the input/output states to account for 4 satellites. The QoE of the model considering all satellites improved by 1.3% and 4.1% on average for 3 users and 5 users, respectively. Our RL design is scalable to support multiple satellites without major changes.

5) *Prioritized ratio strategy*: When total bandwidth is limited, the different resource-sharing strategies may matter. We experiment with two strategies: (i) fair resource sharing and (ii)

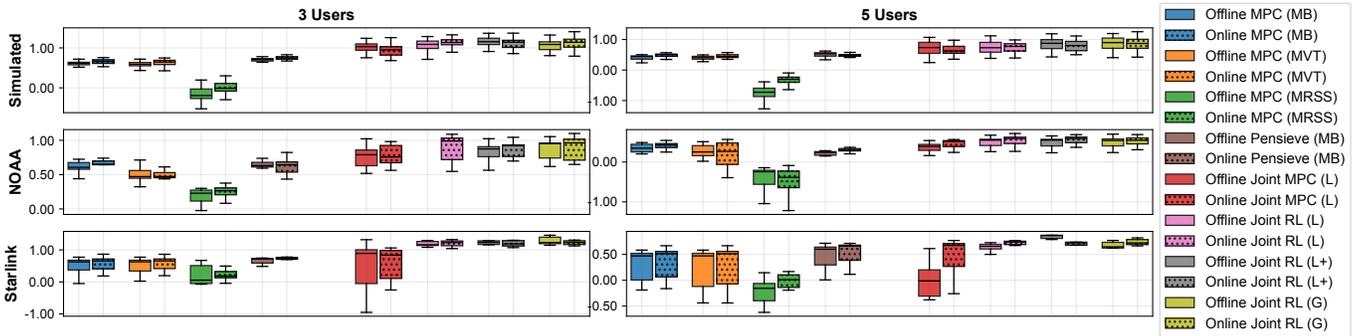


Fig. 12: Comparison of QoE results between testbed and simulation. Testbed results are masked by dots.

TABLE I: QoE results for Joint MPC (Central) with two resource sharing strategies

Sharing Strategy	3 Users	5 Users
Fair Resource	0.999	0.641
Prioritized Ratio	1.032	0.761

TABLE II: QoE comparison among cities (Joint RL (G))

City	Hong Kong	Seoul	Beijing	Chicago	Toronto	Paris	London
Latitude	22.3	37.5	39.9	41.9	43.7	48.9	51.5
QoE - 3 Users	0.87	1.04	0.96	1.10	1.08	1.16	1.31
QoE - 5 Users	0.50	0.71	0.75	0.90	0.94	1.08	1.16

prioritized ratio sharing on the simulated dataset using Joint MPC (Central). We use sequential least squares programming every second to find the optimized resource ratio of users. We filter only those tracks where more than two users are connected to the same satellite. We also apply a heuristic to calculate the rebuffering time by reducing the user’s current buffer size by 70%. This is to accommodate the unexpected events because even a small rebuffering time highly degrades QoE. When we tested using the original buffer sizes in the objective function, the algorithm fully utilized the buffered data; however, this is risky under prediction errors, which can generate rebuffering.

Table I shows that the prioritized ratio strategy outperforms the fair strategy by about 3.3% for 3 users and up to 18.7% for 5 users. It indicates that a prioritized ratio strategy is beneficial when there is a lot of resource competition. Note that this prioritized strategy can only be applied in a centralized model because it should gather the states of all users, such as buffer sizes and previous bitrates, and make decisions for all users.

6) *Multi-session*: We evaluate the three models in multiple sessions, sharing network resources at a single user terminal. We assume that only one session can make a handoff decision to avoid unexpected issues (*e.g.*, each session performing the handoff at different times). Since all sessions share the same user terminal, handoffs must be performed simultaneously. Fig. 11 shows that Joint RL (L) outperforms MPC (MB) and Joint MPC (Central), which is in line with the other evaluations. This result indicates that our proposed models can work well in multi-session problems.

7) *Effects of locations*: According to the data presented in Table II, cities at higher latitudes tend to have higher video QoEs. This is likely because LEO satellites are more concen-

trated in high latitudes, whereas there are fewer satellites in low latitudes [36]. These results come from a simulation. In the real world, other factors such as ground station distribution, obstruction, and user density can also affect the QoE.

D. Testbed Results

To validate the effectiveness of our algorithms in practical scenarios, we implement them in a testbed system as described in §V. As shown in Fig. 12, the results of the online (in testbed) and offline (in simulation) are consistent in most cases. It indicates that joint selection techniques can enhance the video QoE, and Joint RL can outperform other methods.

The major difference we observed in our testbed system is that HTTP requests may get stuck due to network fluctuation and packet loss in an online environment. To address this issue, we have set a timeout threshold of 10 seconds to download each chunk and immediately stop it on the client side if a request exceeds this limit, switch to another satellite, and attempt downloading again. This setting can improve performance, especially on Starlink datasets where obstacles can cause connection timeouts, as shown in Fig. 12. Besides, in the testbed, the latency becomes more unstable than in simulation, and the runtime of each algorithm can also affect performance. We found that our methods are efficient as the bitrate decision of each chunk takes less than 10 *ms*. Thus, the runtime does not adversely impact system performance.

VII. CONCLUSION

We propose a joint satellite selection and adaptive bitrate algorithm based on RL and MPC for LEO satellite networks. According to the experimental results, joint optimization can dynamically optimize satellite selection and bitrate decisions based on network conditions. Our results show that joint selection yields significant benefits in a single-user scenario, and this benefit further increases in a multi-user scenario. Future directions involve leveraging neural network-based video enhancement techniques at receivers to further optimize video streaming in LEO satellite networks (*e.g.*, video super-resolution and interpolation upon packet losses).

VIII. ACKNOWLEDGEMENT

This work is supported in part by NSF Grant CNS-2212297. The authors appreciate the insightful feedback from Yuqing Yang and IEEE INFOCOM 2026 anonymous reviewers.

REFERENCES

- [1] P. Chitre and F. Yegenoglu, "Next-generation satellite networks: architectures and implementations," *IEEE Communications Magazine*, vol. 37, no. 3, pp. 30–36, 1999.
- [2] A. Jamalipour and T. Tung, "The role of satellites in global it: trends and implications," *IEEE Personal Communications*, vol. 8, no. 3, pp. 5–11, 2001.
- [3] H. Zhao, H. Fang, F. Wang, and J. Liu, "Realtime multimedia services over starlink: A reality check," in *Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video*, ser. NOSSDAV '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 43–49. [Online]. Available: <https://doi.org/10.1145/3592473.3592562>
- [4] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 362–373. [Online]. Available: <https://doi.org/10.1145/2018436.2018478>
- [5] K. Park, M. Kim, and L. Park, "Neusaver: Neural adaptive power consumption optimization for mobile video streaming," *IEEE Transactions on Mobile Computing*, vol. 22, no. 11, pp. 6633–6646, 2023.
- [6] Z. He, C. Ge, W. Li, L. Qiu, P. Li, and G. Baig, "Optimized live 4k video multicast streaming on commodity wigig devices," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, 2024, pp. 1131–1142.
- [7] J. Coffey, "Latency in optical fiber systems," 2023, <https://www.commscope.com/globalassets/digizuite/2799-latency-in-optical-fiber-systems-wp-111432-en.pdf>.
- [8] M. Handley, "Delay is not an option: Low latency routing in space," in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, ser. HotNets '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 85–91. [Online]. Available: <https://doi.org/10.1145/3286062.3286075>
- [9] Z. Lai, H. Li, Q. Zhang, Q. Wu, and J. Wu, "Cooperatively constructing cost-effective content distribution networks upon emerging low earth orbit satellites and clouds," in *IEEE 29th International Conference on Network Protocols (ICNP)*, 2021, pp. 1–12.
- [10] Z. Lai, Q. Wu, H. Li, M. Lv, and J. Wu, "Orbitcast: Exploiting mega-constellations for low-latency earth observation," in *IEEE 29th International Conference on Network Protocols (ICNP)*, 2021, pp. 1–12.
- [11] D. Vasisht, J. Shenoy, and R. Chandra, "L2d2: Low latency distributed downlink for leo satellites," in *ACM SIGCOMM*, 2021.
- [12] P.-H. Lin and W. Liao, "Space-centric adaptive video streaming with quality of experience optimization in low earth orbit satellite networks," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 3419–3424.
- [13] 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*, ser. CoNEXT '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 97–108. [Online]. Available: <https://doi.org/10.1145/2413176.2413189>
- [14] 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*, ser. SIGCOMM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 272–285. [Online]. Available: <https://doi.org/10.1145/2934872.2934898>
- [15] K. Spiteri, R. Urganonkar, 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.
- [16] 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*, ser. SIGCOMM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 325–338. [Online]. Available: <https://doi.org/10.1145/2785956.2787486>
- [17] 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*, ser. SIGCOMM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 197–210. [Online]. Available: <https://doi.org/10.1145/3098822.3098843>
- [18] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1928–1937. [Online]. Available: <https://proceedings.mlr.press/v48/mniha16.html>
- [19] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "Qarc: Video quality aware rate control for real-time video streaming based on deep reinforcement learning," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1208–1216. [Online]. Available: <https://doi.org/10.1145/3240508.3240545>
- [20] 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)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 495–511. [Online]. Available: <https://www.usenix.org/conference/nsdi20/presentation/yan>
- [21] X. Zhang, Y. Ou, S. Sen, and J. Jiang, "SENSEI: Aligning video streaming quality with dynamic user sensitivity," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, Apr. 2021, pp. 303–320. [Online]. Available: <https://www.usenix.org/conference/nsdi21/presentation/zhang-xu>
- [22] Y. Guan, Y. Zhang, B. Wang, K. Bian, X. Xiong, and L. Song, "Perm: Neural adaptive video streaming with multi-path transmission," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 1103–1112.
- [23] C. Wang, J. Guan, T. Feng, N. Zhang, and T. Cao, "Bitlat: Bitrate-adaptivity and latency-awareness algorithm for live video streaming," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2642–2646.
- [24] V. Mandawaria, N. Sharma, D. Sharma, C. Majumdar, A. Nigam, S. Park, and J. Jung, "Uplink zone-based scheduling for leo satellite based non-terrestrial networks," in *IEEE WCNC*, 2022.
- [25] B. Yang, Y. Wu, X. Chu, and G. Song, "Seamless handover in software-defined satellite networking," *IEEE Communications Letters*, vol. 20, no. 9, pp. 1768–1771, 2016.
- [26] X. Li, F. Tang, L. Chen, and J. Li, "A state-aware and load-balanced routing model for leo satellite networks," in *IEEE GLOBECOM*, 2017.
- [27] I. F. Akyildiz, H. Uzunali, and M. D. Bender, "Handover management in low earth orbit (leo) satellite networks," in *Mobile Networks and Applications*, 1999.
- [28] H. N. Nguyen, S. Lepaja, J. Schuringa, and H. van As, "Handover management in low earth orbit satellite ip networks," in *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No.01CH37270)*, vol. 4, 2001, pp. 2730–2734 vol.4.
- [29] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Handover solutions for 5g low-earth orbit satellite networks," *IEEE Access*, vol. 10, pp. 93 309–93 325, 2022.
- [30] "Starlink Simulator," 2023, <https://starlink.sx>.
- [31] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Favor: fine-grained video rate adaptation," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 64–75. [Online]. Available: <https://doi.org/10.1145/3204949.3204957>
- [32] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.
- [33] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *arXiv:1710.06542*, 2017.
- [34] C. Yu, A. Velu, E. Vinitzky, J. Gao, Y. Wang, A. Bayen, and Y. WU, "The surprising effectiveness of ppo in cooperative multi-agent games," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 24 611–24 624.
- [35] S. Schneider, H. Karl, R. Khalili, and A. Hecker, "Deepcomp: Coordinated multipoint using multi-agent deep reinforcement learning," 2021.
- [36] J. C. McDowell, "The low earth orbit satellite population and impacts of the spacex starlink constellation," in *The Astrophysical Journal Letters*, 2020.