

DSROQ: Dynamic Scheduling and Routing for QoE Management in LEO Satellite Networks

Dhiraj Bhattacharjee*, Pablo G. Madoery*, Abhishek Naik[†], Halim Yanikomeroglu*,
Gunes Karabulut Kurt[‡], Stéphane Martel[§], Khaled Ahmed[§]

* Department of Systems and Computer Engineering, Carleton University, Canada

[†] National Research Council Canada, Canada

[‡] Department of Electrical Engineering, Polytechnique Montréal, Canada

[§] Satellite Systems, MDA, Canada

Abstract—The modern Internet supports diverse applications with heterogeneous quality of service (QoS) requirements. Low Earth orbit (LEO) satellite constellations offer a promising solution to meet these needs, enhancing coverage in rural areas and complementing terrestrial networks in urban regions. Ensuring QoS in such networks requires joint optimization of routing, bandwidth allocation, and dynamic queue scheduling, as traffic handling is critical for maintaining service performance. This paper formulates a joint routing and bandwidth allocation problem where QoS requirements are treated as soft constraints, aiming to maximize user experience. An adaptive scheduling approach is introduced to prioritize flow-specific QoS needs. We propose a Monte Carlo tree search (MCTS)-inspired method to solve the NP-hard route and bandwidth allocation problem, with Lyapunov optimization-based scheduling applied during reward evaluation. Using the Starlink Phase 1 Version 2 constellation, we compare end-user experience and fairness between our proposed DSROQ algorithm and a benchmark scheme. Results show that DSROQ improves both performance metrics and demonstrates the advantage of joint routing and bandwidth decisions. Furthermore, we observe that the dominant performance factor shifts from scheduling to routing and bandwidth allocation as traffic sensitivity changes from latency-driven to bandwidth-driven.

Index Terms—Fairness, Lyapunov optimization, low Earth orbit, Monte Carlo tree search, QoE, QoS, routing, satellite mega-constellations, scheduling.

I. INTRODUCTION

The global push for universal Internet access has intensified with the rise of 6G, which promises ultra-fast, reliable, and versatile communication. A key challenge is bridging the digital divide—the gap between those with and without access to high-speed internet. Satellites are vital to this effort, especially in remote regions lacking terrestrial infrastructure, and in urban areas with high demand. Non-terrestrial networks (NTNs), led by industry players like SpaceX, Telesat, Amazon, and OneWeb, are gaining momentum in both industry and academia, in particular, low Earth orbit (LEO) satellite constellations due to their lower latency and higher data rates.

Today’s Internet supports diverse applications from VoIP and video conferencing (VC) to live streaming (LS), messaging, and file transfers (FT), each with distinct quality of service (QoS) needs defined by latency, throughput, packet loss, and jitter. While QoS refers to measurable network performance metrics, quality of experience (QoE) is the user-perceived performance. Moreover, flows from different applications may

carry varying priorities based on service level agreements (SLAs). Meeting these heterogeneous QoS and SLA requirements to optimize QoE depends on two critical decisions: (i) routing and bandwidth allocation at the network level, and (ii) queue management at the node level. Optimizing either in isolation may fall short. Thus, a joint approach that combines both is crucial for delivering optimal user experience.

Early queue management techniques like first-in-first-out (FIFO), priority queuing (PQ), and weighted fair queuing (WFQ) were designed to provide simplicity, prioritization for critical traffic, and fair bandwidth allocation, respectively. Building on these, integrated services (IntServ) [1] introduced per-flow resource reservations for strict QoS but lacked scalability, prompting the development of differentiated services (DiffServ) [2], which classifies traffic as expedited forwarding (EF), assured forwarding (AF), and best effort (BE). Active queue management (AQM) techniques emerged to mitigate congestion proactively. Random early detection (RED) [3] drops packets probabilistically to avoid buffer overflows but required careful tuning whereas controlled delay (CoDel) [4] improves latency by dropping packets exceeding a delay threshold with minimal configuration. Flow queue CoDel (FQ-CoDel) [5] improves fairness and delay control by combining CoDel with per-flow queuing, making it effective for diverse and modern traffic environments.

Effective queue management enhances traffic handling within the network, but optimal routing strategies are equally crucial for meeting diverse QoS requirements and maximizing overall performance. In [6], a multi-level priority queuing scheme is proposed where priorities adapt dynamically based on delay estimation and QoS violations. [7] builds on this by combining priority queuing with weighted round-robin (WRR) scheduling for three traffic classes. [8] uses bandwidth and delay metrics for route selection with FIFO queues. [9] sequentially computes routes for delay-sensitive, bandwidth-sensitive, and BE flows using Dijkstra’s algorithm for minimum latency, maximum bandwidth, and fewest hops, respectively. Finally, [10] proposes fybrrLink, which uses flow-specific weights (latency, bandwidth, reliability) to compute a modified-cost shortest path via a constrained search space generated using a modified Bresenham’s algorithm.

While extensive research exists on QoS-aware routing in

satellite networks, most works do not assume specific queuing architectures despite queuing delay being a key factor influencing routing decisions. Some studies incorporate priority queuing or combine it with WRR schedulers, but the absence of joint routing and node-level scheduling decisions often leads to suboptimal resource utilization. In parallel, the combinatorial complexity of joint routing across large-scale flows has led many recent approaches to adopt sequential methods, which compromise overall network performance. Moreover, modern queue management techniques primarily emphasize per-hop behavior, ignoring routing context and end-to-end network conditions. Traditional models also treat QoS constraints rigidly, deeming any violation infeasible. A more flexible strategy considers QoS as soft objectives, aiming to minimize violations or maximize satisfaction. To the best of our knowledge, no prior work jointly optimizes routing, bandwidth allocation, and queue scheduling while accounting for diverse traffic priorities and QoS requirements. We address this gap by proposing a unified optimization framework across all flows that maximizes the weighted sum of end-user experience.

The paper organization is as follows. Section II introduces the network architecture. Section III formulates the joint routing and bandwidth allocation problem along with dynamic queue scheduling problem to maximize overall user experience. Section IV presents a tractable solution using Monte Carlo tree search (MCTS) for routing and bandwidth allocation, combined with a Lyapunov optimization-based queue scheduling algorithm. Section V compares the proposed approach with a benchmark scheme. Finally, Section VI concludes the paper and outlines directions for future work.

II. NETWORK ARCHITECTURE

We consider a LEO satellite network with regenerative payloads, where satellites are interconnected with up to four bidirectional inter-satellite links (ISLs) to their immediate neighbors. The network is represented as a graph $(\mathcal{V}, \mathcal{E})$, where \mathcal{V} and \mathcal{E} denote the sets of satellites and ISLs, respectively. All ISLs have the same unidirectional link capacity, R_l . Ground traffic generation and reception are abstracted by modeling both source and destination of a flow f as satellites, i.e., the satellite serving a ground area acts as the source or destination of flows from/to that area. Each flow is identified by a unique ID, source and destination satellite, and application type. To improve scalability and reduce per-flow control complexity, β flows with the same source, destination, and application type are aggregated into a single aggregated flow f . To capture the dynamics of mobile LEO satellites, we adopt the virtual node model from [11], where each satellite is associated with a fixed ground area it serves. As the topology evolves, the next satellite in orbit assumes responsibility for that area. To enable seamless transitions, routing information is transferred to the incoming satellite prior to handover, ensuring continuity in routing decisions.

Each aggregated flow has QoS requirements defined by upper and lower bounds on latency $(\delta_f^{\max}, \delta_f^{\min})$, through-

put $(\tau_f^{\max}, \tau_f^{\min})$, and packet drop rate (l_f^{\max}, l_f^{\min}) , which influence the end-user experience. Let \mathcal{F} denote the set of F aggregated flows, partitioned into EF, AF, and BE classes: $\mathcal{F} = \mathcal{F}_{EF} \cup \mathcal{F}_{AF} \cup \mathcal{F}_{BE}$. Each aggregated flow is assigned an SLA-based weight w_f . We define the time slot t as the time needed to transmit a packet, i.e., packet length divided by link capacity. Packets arriving via ISLs or generated at a satellite are processed by a routing module. If the satellite is not the destination, the routing module determines the next hop, and the packet is forwarded to one of four queuing modules. Each queuing module classifies packets by their aggregated flow ID and stores them in dedicated per-aggregated-flow queues. A scheduler then selects one aggregated flow to transmit its head-of-queue (HoQ) packet over the link. The scheduling index $s_{f,\{i,j\}}^t$ is a binary variable set to 1 if the aggregated flow f is scheduled at time slot t on the outbound ISL $\{i,j\}$ from satellite i to j , and 0 otherwise. For packet dropping due to buffer overflow, packets are considered to be dropped in round robin fashion from the aggregated flows which share a common outbound link.

III. PROBLEM FORMULATION

To evaluate the performance of each aggregated flow f at its destination, we measure average latency (δ_f^T) , throughput (τ_f^T) , and packet drop rate (l_f^T) over a time window T comprising $Z > 1$ time slots. Based on these metrics, we compute a QoS score reflecting user experience. The best and worst QoS scores are denoted by Ω^{\max} and Ω^{\min} , respectively. Latency-based QoS score Ω_f^δ is assigned as Ω^{\max} if $\delta_f^T \leq \delta_f^{\min}$, and Ω^{\min} if $\delta_f^T \geq \delta_f^{\max}$. For intermediate values, Ω_f^δ is linearly interpolated between Ω^{\max} and Ω^{\min} . A similar mapping is applied to the average packet drop rate l_f^T to compute the QoS score Ω_f^l , using the bounds l_f^{\min} and l_f^{\max} . For throughput, the QoS score Ω_f^τ is computed inversely to latency: Ω^{\max} is assigned when $\tau_f^T \geq \tau_f^{\max}$, and Ω^{\min} when $\tau_f^T \leq \tau_f^{\min}$. For intermediate values, Ω_f^τ is linearly interpolated between Ω^{\min} and Ω^{\max} . The final QoS score Ω_f^T is computed as a weighted sum of the individual scores: $\Omega_f^T = \zeta_f^\delta \Omega_f^\delta + \zeta_f^\tau \Omega_f^\tau + \zeta_f^l \Omega_f^l$, where ζ_f^δ , ζ_f^τ , and ζ_f^l are the respective weights for latency, throughput, and packet drop rate for the aggregated flow f .

A. Route and Bandwidth Allocation

Let K_f denote the number of available routes for aggregated flow f from source to destination. We define a binary vector α_f^T , where each element $\alpha_{f,r}^T \in \{0, 1\}$ indicates whether route r is selected for aggregated flow f during time window T . The selected route for aggregated flow f with $\alpha_{f,r}^T = 1$ is denoted by r_f^T , representing an ordered list of edges from source to destination. Upon route selection, the allocated bandwidth B_f^T must be determined as well. The bandwidth B_f^T is chosen from a discrete set \mathcal{B}_f of B_w values uniformly spanning $[\tau_f^{\min}, \tau_f^{\max}]$, including both endpoints. Our objective is to maximize the weighted sum of the QoS scores of all the aggregated flows $f \in \mathcal{F}$ scaled to 1 as presented below:

$$\begin{aligned}
\max_{\alpha_{f,r}^T, B_f^T} \quad & \frac{\sum_{f \in \mathcal{F}} w_f \Omega_f^T}{\Omega^{\max} \sum_{f \in \mathcal{F}} w_f} \quad (1a) \\
\text{s.t.} \quad & \sum_{r=1}^{K_f} \alpha_{f,r}^T = 1, \forall f \in \mathcal{F}, \quad (1b) \\
& \alpha_{f,r}^T \in \{0, 1\}, \forall f \in \mathcal{F}, r = 1, 2, \dots, K_f, \quad (1c) \\
& B_f^T \in \mathcal{B}_f, \forall f \in \mathcal{F}, \quad (1d) \\
& \sum_{f: \{i,j\} \in r_f^T} B_f^T \leq R_l, \forall \{i,j\} \in \mathcal{E}. \quad (1e)
\end{aligned}$$

The maximization in (1a) selects optimal routes $\alpha_{f,r}^T$ and bandwidths B_f^T for all aggregated flows. Constraint (1b) ensures each flow selects one route among K_f options, while (1c) enforces the binary nature of $\alpha_{f,r}^T$. (1d) restricts B_f^T to discrete values in \mathcal{B}_f , and (1e) ensures that the total bandwidth on any link does not exceed its capacity R_l . The problem is an integer non-linear program (INLP) due to the non-linear objective, and is NP-hard [12]. To address this with tractable complexity, we propose a scalable MCTS-based joint routing and bandwidth allocation scheme, detailed in the next section.

B. Queue Scheduling

With the given routes and bandwidth allocations for all aggregated flows, the scheduler must determine, at each time slot, which aggregated flow to serve from the set of aggregated flows traversing the link $\{i, j\}$. Since throughput can be evaluated on a per-hop basis, we estimate the latency-based QoS score at each hop and each time slot, denoted by $\Omega_{f,\{i,j\}}^{\delta,t}$, as follows:

$$\Omega_{f,\{i,j\}}^{\delta,t} = \Omega^\delta(\delta_{f,\{i,j\}}^t, \delta_{f,\{i,j\}}^{\max}, \delta_{f,\{i,j\}}^{\min}), \quad (2)$$

where $\Omega^\delta(\cdot)$ denotes the delay-to-QoS mapping function. Here, $\delta_{f,\{i,j\}}^t$ represents the local queuing delay of the HoQ packet at satellite i . The parameters $\delta_{f,\{i,j\}}^{\max}$ and $\delta_{f,\{i,j\}}^{\min}$ are modified delay bounds used to estimate the latency-based QoS score. These modified bounds are obtained by subtracting, from the original end-to-end delay bounds, the sum of the propagation delay, $\delta_{f,\{i,j\}}^{\text{prop}}$ and transmission delay, $\delta_{f,\{i,j\}}^{\text{trans}}$ of the selected route r_f^T scaled by the number of hops $|r_f^T|$. The resulting per-hop queuing delay bound is further adjusted by the congestion factor $\gamma_{\{i,j\}}/\bar{\gamma}_f$, where $\gamma_{\{i,j\}}$ denotes the average link utilization of ISL $\{i, j\}$ and $\bar{\gamma}_f$ is the average link utilization over the entire route of aggregated flow f . For example, the modified upper latency bound is given by:

$$\delta_{f,\{i,j\}}^{\max} = \frac{\delta_f^{\max} - \sum_{\{i,j\} \in r_f^T} (\delta_{f,\{i,j\}}^{\text{prop}} + \delta_{f,\{i,j\}}^{\text{trans}})}{|r_f^T|} \cdot \frac{\gamma_{\{i,j\}}}{\bar{\gamma}_f}. \quad (3)$$

In (3), the inclusion of the congestion factor adjusts the bound relative to the route-wide average link utilization of the aggregated flow's route, stretching it for links whose average link utilization exceeds this route-wide average and compressing it for links whose utilization is lower, thereby accounting

for localized congestion in scheduling decisions. The weighted estimated latency-based QoS score can be written as,

$$g_{f,\{i,j\}}^t = w_f \zeta_f^\delta \Omega^\delta(\delta_{f,\{i,j\}}^t + 1 - s_{f,\{i,j\}}^t, \delta_{f,\{i,j\}}^{\max}, \delta_{f,\{i,j\}}^{\min}) \quad (4)$$

(4) represents that the local HoQ delay is in terms of time slots and gets incremented by one if not scheduled. To represent the throughput deficit of an aggregated flow, we define a concatenated vector of virtual queues, $\mathbf{V}_{\{i,j\}}^t = [V_{f,\{i,j\}}^t]; \forall f: \{i,j\} \in r_f^T$ and each virtual queue is updated as follows:

$$V_{f,\{i,j\}}^{t+1} = \max[V_{f,\{i,j\}}^t + B_f^T - s_{f,\{i,j\}}^t, 0]. \quad (5)$$

With this, we formulate the following scheduling problem for each outbound ISL $\{i, j\}$ considering N time slots:

$$\max_{s_{f,\{i,j\}}^t} \quad \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \sum_{f: \{i,j\} \in r_f^T} g_{f,\{i,j\}}^t \quad (6a)$$

$$\text{s.t.} \quad \lim_{N \rightarrow \infty} \frac{\mathbb{E}[V_{f,\{i,j\}}^t]}{N} = 0, \forall f: \{i,j\} \in r_f^T, \forall t, \quad (6b)$$

$$s_{f,\{i,j\}}^t \in \{0, 1\}, \forall f: \{i,j\} \in r_f^T, \forall t, \quad (6c)$$

$$\sum_{f: \{i,j\} \in r_f^T} s_{f,\{i,j\}}^t = 1, \forall t. \quad (6d)$$

The problem (6a) determines the optimal scheduling index, $s_{f,\{i,j\}}^t$, $\forall t$ for the aggregated flows with routes passing through ISL $\{i, j\}$. Constraint (6b) ensures mean rate stability of $V_{f,\{i,j\}}^t$, i.e., ensuring the throughput to be stabilized around the assigned bandwidth, B_f^T . Finally, (6c) and (6d) ensures the binary nature of the scheduling index and scheduling only one aggregated flow per time slot, respectively.

IV. DYNAMIC SCHEDULING AND ROUTING FOR QOE MANAGEMENT (DSROQ)

A. MCTS-based Routing and Bandwidth Allocation

MCTS, introduced by Coulom [13], is a popular algorithm for solving combinatorial and sequential decision problems, especially in large and uncertain state spaces. MCTS builds a search tree through four iterative steps: selection (choosing nodes by balancing exploration and exploitation), expansion (adding new child nodes), simulation (estimating outcomes via random rollouts), and backpropagation (updating node values based on simulation results). This iterative process enables efficient discovery of high-reward solutions. A detailed description of these steps follows.

1) *Tree Structure*: Each training episode starts from an empty network at the root node $S_0 = \{\}$. At each tree level, beginning with S_0 , a configuration of route and bandwidth, $\{\alpha_f^T, B_f^T\}$ is selected for an aggregated flow f and appended to the current node to form a child node. For example, after selecting $\{\alpha_1^T, B_1^T\}$ at level 1 for aggregated flow 1, the transition $S_0 \rightarrow S_{11}$ yields $S_{11} = \{\alpha_1^T, B_1^T\}$. At level 2, choosing $\{\alpha_2^T, B_2^T\}$ leads to $S_{11} \rightarrow S_{21}$, where $S_{21} = \{\{\alpha_1^T, B_1^T\}, \{\alpha_2^T, B_2^T\}\}$. Continuing this process across flows and levels constructs the complete route and bandwidth allocations upon reaching a leaf node.

2) *Selection, Expansion, and Simulation*: Starting from S_0 , if the current node has no children, a random rollout is performed, selecting random configurations until reaching a leaf node. Otherwise, it applies an ϵ -greedy strategy to either select an existing child node or expand a new one.

3) *Reward*: At the end of each episode, upon reaching a leaf node with the full configuration $\prod_{f \in \mathcal{F}} \alpha_f^T, B_f^T$, the objective function (1a) is computed based on the QoS scores. To account for constraint violation, a cost function C is defined for (1e), while constraints (1b)–(1d) are inherently satisfied through discrete decision variables, as shown below:

$$C = \sum_{\{i,j\} \in \mathcal{E}} \frac{\max(0, \sum_{f: \{i,j\} \in r_f^T} B_f^T - R_l)}{R_l}. \quad (7)$$

Here, C penalizes bandwidth allocations exceeding link capacity R_l . The final reward can be written as,

$$\mathcal{R} = \frac{\sum_{f \in \mathcal{F}} w_f \Omega_f^T}{\Omega^{\max} \sum_{f \in \mathcal{F}} w_f} - \lambda C, \quad (8)$$

where λ balances the trade-off between objective maximization and constraint satisfaction.

4) *Backpropagation*: Each node maintains a Q-value representing its selection quality. After completing an episode, the computed reward \mathcal{R} from (8) is backpropagated from the leaf node to the root through the connecting edges. At each node S , the Q-value is updated as,

$$Q(S) \leftarrow \max\{Q(S), \mathcal{R}\}. \quad (9)$$

Finally, the ϵ value is updated at each episode using the following logarithmic function of the number of episodes, starting at $(1 - \epsilon_0)$ and gradually decreasing to ϵ^{\min} :

$$\epsilon \Big|_{episode=z} = \max\{1 - \epsilon_0 * \log_{10}(\frac{z}{a_0} + b_0), \epsilon^{\min}\}. \quad (10)$$

Over multiple episodes, the MCTS policy learns to identify the near-optimal path from the root node to one of the leaf nodes yielding the highest reward encountered during training.

B. Lyapunov Optimization based Dynamic Scheduling

In (6a), to handle the stochastic constraint (6b), using Lyapunov optimization, the Lyapunov function of the concatenated vectors are defined as,

$$L[\mathbf{V}_{\{i,j\}}^t] \triangleq \frac{1}{2} \sum_{f: \{i,j\} \in r_f^T} \left(V_{f,\{i,j\}}^t \right)^2, \forall t, \quad (11)$$

while the conditional Lyapunov drift is defined as,

$$\Delta(\mathbf{V}_{\{i,j\}}^t) \triangleq \mathbb{E} \left[L[\mathbf{V}_{\{i,j\}}^{t+1}] - L[\mathbf{V}_{\{i,j\}}^t] \middle| \mathbf{V}_{\{i,j\}}^t \right]. \quad (12)$$

From [14], the upper bound of $\Delta(\mathbf{V}_{\{i,j\}}^t)$ can be written as,

$$\Delta(\mathbf{V}_{\{i,j\}}^t) \leq B + \sum_{f: \{i,j\} \in r_f^T} \mathbb{E} \left[V_{f,\{i,j\}}^t (B_f^T - s_{f,\{i,j\}}^t) \middle| \mathbf{V}_{\{i,j\}}^t \right], \quad (13)$$

where B is a scalar constant. The overall objective is maximizing (6a) while minimizing the upper bound of $\Delta(\mathbf{V}_{\{i,j\}}^t)$, which leads to the following objective function:

$$G_{\{i,j\}}^t = \sum_{f: \{i,j\} \in r_f^T} g_{f,\{i,j\}}^t - w_f \zeta_f^T \Delta(\mathbf{V}_{\{i,j\}}^t). \quad (14)$$

Now, putting the upper bound of the conditional Lyapunov drift in (14), and only keeping the relevant terms, the modified objective is presented as follows:

$$G_{\{i,j\}}^{t'} = \sum_{f: \{i,j\} \in r_f^T} w_f \left\{ \zeta_f^\delta \Omega^\delta \left(\delta_{f,\{i,j\}}^t + 1 - s_{f,\{i,j\}}^t \right) + \zeta_f^T V_{f,\{i,j\}}^t s_{f,\{i,j\}}^t \right\}. \quad (15)$$

After mathematical manipulation, (15) can be solved with constraints (6c) and (6d), and the optimal aggregated flow f^* at the egress port of ISL $\{i,j\}$ at time slot t with $s_{f,\{i,j\}}^t = 1$ can be written as follows:

$$f^* = \arg \max_{f: \{i,j\} \in r_f^T} w_f \left\{ \zeta_f^\delta \left[\Omega^\delta \left(\delta_{f,\{i,j\}}^t \right) - \Omega^\delta \left(\delta_{f,\{i,j\}}^t + 1 \right) \right] + \zeta_f^T V_{f,\{i,j\}}^t \right\}. \quad (16)$$

DSROQ integrates the MCTS-based policy for route and bandwidth allocation with the scheduling policy defined in (16). During each MCTS training episode, when a leaf node is reached, the reward is evaluated using (8), based on the average network performance over a time window of Z time slots. Within this window, and for every time slot, the packet to be transmitted is selected using the scheduling policy (16). Thus, the scheduling policy directly influences the reward estimation in MCTS, creating a tightly coupled loop between long-term path decisions and short-term queue management.

V. SIMULATION RESULTS

A. Simulation Settings

In this section, we present and compare the training and testing results with fybrLink [10], which calculates routes sequentially for each incoming flow. We evaluate fybrLink under a prioritized scheme where EF flows are served first, followed by AF and BE. Two algorithm variants are tested: DSROQ and DSROQ-FIFO. Both use MCTS-learned route and bandwidth allocations; DSROQ applies the scheduling rule in (16), while DSROQ-FIFO uses a FIFO queue per egress port toward each outbound ISL. We calculate propagation delay for intra and inter-orbital ISLs in a 4-by-4 satellite configuration, using the Starlink Phase 1 Version 2 constellation (550 km altitude, 53° inclination). The simulation runs for 60 seconds with topology changes every 15 seconds. We simulate 10,000 flows across VC (EF), LS (AF), video-on-demand (AF), and FT (BE) applications with distributions of 20%, 20%, 20%, and 40%, respectively for 100 iterations. Packet generation for each flow is modeled as a Poisson random variable with a mean equal to its allocated bandwidth. EF, AF and BE flow weights are considered as 20, 2 and 1, respectively unless stated otherwise. The latency, bandwidth, and packet

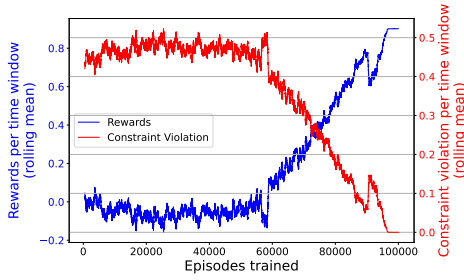


Fig. 1. Reward and constraint violation vs episodes trained.

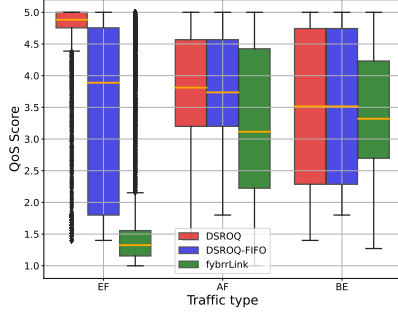


Fig. 2. Comparison of QoS scores.

loss related QoS bounds are considered from [15], [16], and [17], respectively. In addition, $[\zeta_f^\delta, \zeta_f^\tau, \zeta_f^l]$ for EF, AF, and BE flows are considered as $[0.8, 0.1, 0.1]$, $[0.1, 0.45, 0.45]$, and $[0.1, 0.8, 0.1]$, respectively [18]. Finally, Ω_{max} and Ω_{min} are considered as 5 and 1, respectively.

B. Reward and Constraint Violation

Fig. 1 shows the rolling mean (window size 500) of reward and constraint violations per time window versus training episodes. Reward values increase and eventually saturate as training progresses, driven by the decay of the exploration factor ϵ toward $\epsilon^{\min} = 0$ as per (10). Similarly, constraint violations decrease and converge to zero, indicating successful training and constraint compliance.

C. User Experience

Fig. 2 compares the QoS scores of the proposed DSROQ algorithm against the benchmark for EF, AF, and BE traffic classes. The box plots show that the median QoS score (yellow lines) decreases across DSROQ variants from EF to AF to BE flows, reflecting the priority weights used during training and testing ($EF > AF > BE$). In contrast, fybrrLink exhibits an increasing median QoS trend from EF to BE flows, as QoS requirements become progressively relaxed. Compared to the benchmark, DSROQ achieves significant QoS improvements for EF and AF flows, with the improvement diminishing toward BE flows. Although several outliers are visible in the DSROQ EF scores, they represent around 4% of the total data and do not significantly affect the overall distribution. Comparing DSROQ and DSROQ-FIFO reveals a large difference for EF flows in terms of median QoS scores and inter-quartile ranges, which reduces for AF and becomes negligible for BE. First of all, changing the scheduling policy from (16) to a FIFO rule removes the priority in scheduling EF packets and gives

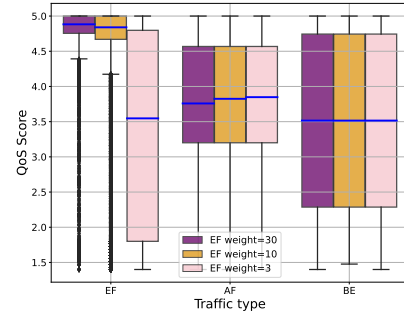


Fig. 3. Effect on varying priority weights on QoS scores.

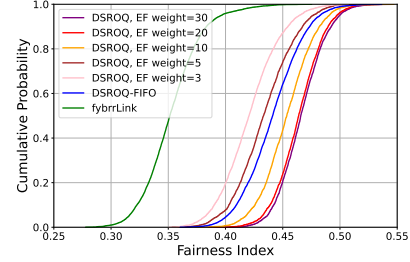


Fig. 4. CDF of fairness.

equal chance to an AF or BE packet to get scheduled. This leads to a latency increase in EF flows but latency reduction in AF and BE flows. Secondly, when the average incoming traffic is below link capacity, scheduling mainly impacts delay rather than throughput. Since EF flows are highly delay-sensitive ($\zeta_f^\delta/\zeta_f^\tau = 8$), change in scheduling policy has a strong effect in overall QoS distribution for EF flows, whereas for AF ($\zeta_f^\delta/\zeta_f^\tau = 0.22$) and BE ($\zeta_f^\delta/\zeta_f^\tau = 0.125$) flows, the impact is progressively smaller.

D. Effect of Priority Weights

Fig. 3 shows the QoS score variation across three EF priority weights, with route and bandwidth allocations trained for an EF weight of 20 (AF and BE weights of 2 and 1, respectively used in both training and testing). For high EF weights, the median QoS scores (blue lines) follow $EF > AF > BE$. However, when the EF weight is lesser, this order changes. Reducing the EF weight lowers the scheduling priority of EF packets, increasing their latency and degrading QoS for latency-sensitive EF flows. This reduces median QoS scores for EF flows along with increase in outlier percentage from 1.6% to 14.3% for EF weight of 20 and 10. Meanwhile, AF and BE packets are scheduled more often, slightly improving AF QoS (due to moderate latency sensitivity) and leaving BE QoS largely unchanged (due to minimal latency sensitivity). These results highlight that throughput is mainly determined by route and bandwidth allocations, while latency is influenced by scheduling, route, and bandwidth allocations.

E. Fairness Index

Fig. 4 presents a comparative analysis of the cumulative distribution function (CDF) of fairness index between the benchmark algorithm and proposed DSROQ under different

EF priority weights using the route and bandwidth allocations trained with an EF weight of 20 (AF and BE weights fixed at 2 and 1, respectively). Fairness of the QoS scores provided to each aggregated flow is measured using the QoE fairness index from [19]. We observe that DSROQ consistently achieves higher fairness than fybrrLink, primarily due to its joint route and bandwidth allocation that considers all flows simultaneously, as opposed to fybrrLink's sequential processing. In addition, DSROQ with EF weight of 20 also outperforms DSROQ-FIFO in fairness, which aligns with Fig. 2, where the inter-quartile range of EF QoS scores increases from DSROQ to DSROQ-FIFO, indicating higher QoS variance and lower fairness. Furthermore, increasing the EF weight improves fairness among flows in DSROQ. This is supported by Fig. 3, which shows that higher EF weights reduce the spread in EF QoS scores, thereby increasing fairness.

VI. CONCLUSION AND FUTURE WORK

Meeting heterogeneous QoS requirements and service-level priorities in satellite networks relies on routing, bandwidth allocation, and scheduling decisions. This paper models end-user experience as QoS scores derived from network performance metrics, and formulates the network performance as a weighted sum of QoS scores, where weights reflect service-level priorities. To maximize this metric, we pose an optimization problem for jointly selecting routes and bandwidth allocations, along with an adaptive scheduling problem that accounts for flow-specific QoS needs and priorities. An MCTS-inspired approach is proposed to address the NP-hard problem of route and bandwidth allocation, with Lyapunov optimization-based scheduling solution used during reward evaluation in MCTS. The resulting algorithm, DSROQ, combines both the MCTS-based resource allocation and the proposed scheduler. We evaluate two DSROQ variants against the benchmark fybrrLink in a 4-by-4 satellite grid within the Starlink Phase 1 Version 2 constellation. Both DSROQ variants outperform fybrrLink in QoS score and fairness. DSROQ-FIFO's improvement over fybrrLink highlights the benefit of joint routing and bandwidth allocation across all flows, while DSROQ's advantage over DSROQ-FIFO emphasizes the critical role of scheduling, especially for latency-sensitive EF flows. This effect is further validated by varying EF priority weights, where the dominant influence shifts from scheduling to routing and bandwidth allocation as flow sensitivity moves from latency to bandwidth.

While this paper optimizes the network over a fixed time window for multiple flows, future extensions may consider dynamic networks where traffic arrives and departs randomly. In such settings, resource allocation could either react to available capacity on arrival or perform joint optimization periodically to accommodate all active flows. In future, we plan to extend this framework to support joint admission control, routing, bandwidth allocation, and scheduling for such dynamic and diverse traffic scenarios.

REFERENCES

- [1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," IETF, Fremont, CA, USA, Tech. Rep. RFC 1633, Jul. 1994.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," IETF, Fremont, CA, USA, RFC 2475, Dec. 1998, [Online]. Available: <https://www.rfc-editor.org/info/rfc2475>.
- [3] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1(4), pp. 397–413, Aug. 1993.
- [4] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, "Controlled delay active queue management," IETF, Fremont, CA, USA, RFC 8289, Jan. 2018.
- [5] T. Hoeiland-Joergensen, P. McKeeney, D. Taht, J. Gettys, and E. Dumazet, "The flow queue codel packet scheduler and active queue management algorithm," IETF, Fremont, CA, USA, RFC 8290, Jan. 2018.
- [6] C. Hou and Y. Zhu, "The QoS guaranteed routing strategy in low Earth orbit satellite constellations," in *Proc. IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, Dalian, China, 2023, pp. 1–6.
- [7] J. Zhang, R. Ding, J. Liu, R. Zhang, and H. Lu, "QoSRA: a QoS-aware routing algorithm for software defined satellite networks," in *Proc. 2nd Information Communication Technologies Conference (ICTC)*, Nanjing, China, 2021, pp. 165–171.
- [8] Q. Liu, X. Li, H. Ji, and H. Zhang, "Multi-path routing algorithm with joint optimization of load-balancing for cluster-based LEO satellite networks," in *Proc. 8th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*, Beijing, China, 2023, pp. 264–268.
- [9] Y. Huang, B. Feng, A. Tian, P. Dong, S. Yu, and H. Zhang, "An efficient differentiated routing scheme for MEO/LEO-based multi-layer satellite networks," *IEEE Transactions on Network Science and Engineering*, vol. 11(1), pp. 1026–1041, Jan.-Feb. 2024.
- [10] P. Kumar, S. Bhushan, D. Halder, and A. M. Baswade, "fybrrLink: efficient QoS-aware routing in SDN enabled future satellite networks," *IEEE Transactions on Network and Service Management*, vol. 19(3), pp. 2107–2118, Sep. 2022.
- [11] R. Mauger and C. Rosenberg, "QoS guarantees for multimedia services on a TDMA-based satellite network," *IEEE Communications Magazine*, vol. 35(7), pp. 56–65, Jul. 1997.
- [12] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*. New York: W.H. Freeman and Company, 1979.
- [13] R. Coulom, "Efficient selectivity and backup operators in monte-carlo tree search," in *International conference on computers and games*. Springer, Berlin, Heidelberg, 2006, pp. 72–83.
- [14] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers, 2010.
- [15] Cisco Systems, "Quality of service design overview," [Online]. Available: <https://www.ciscopress.com/articles/article.asp?p=357102&seqNum=7>, accessed on Feb. 07, 2025.
- [16] FCC, "Broadband speed guide," [Online]. Available: <https://www.fcc.gov/consumers/guides/broadband-speed-guide>, accessed on Feb. 07, 2025.
- [17] A. Al-Jawad, I.-S. Comşa, P. Shah, O. Gemikonakli, and R. Trestian, "An innovative reinforcement learning-based framework for quality of service provisioning over multimedia-based SDN environments," *IEEE Transactions on Broadcasting*, vol. 67(4), pp. 851–867, Dec. 2021.
- [18] B. Mao, X. Zhou, J. Liu, and N. Kato, "On an intelligent hierarchical routing strategy for ultra-dense free space optical low Earth orbit satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 42(5), pp. 1219–1230, May 2024.
- [19] T. Hoßfeld, L. Skorin-Kapov, P. E. Heegaard, and M. Varela, "Definition of QoE fairness in shared systems," *IEEE Communications Letters*, vol. 21(1), pp. 184–187, Jan. 2017.