Reinforcement Learning for QoE-Oriented Flexible Bandwidth Allocation in Satellite Communication Networks

Teweldebrhan Mezgebo Kebedew
SnT, University of Luxembourg
Teweldebrhan.Kebedew@uni.lu

Vu Nguyen Ha
SnT, University of Luxembourg
vu-nguyen.ha@uni.lu

Eva Lagunas
SnT, University of Luxembourg
eva.lagunas@uni.lu

Duc Dung Tran
SnT, University of Luxembourg
duc.tran@uni.lu

Joel Grotz
SES S.A., Luxembourg
joel.grotz@ses.com

Symeon Chatzinotas
SnT, University of Luxembourg
symeon.chatzinotas@uni.lu

***Abstract*—Optimizing the use of satellite bandwidth to achieve maximum return in a system where user demands are constantly changing, and application-specific Quality-of-Experience (QoE) requirements need to be met, presents a complex challenge for both satellite operators and service providers (SPs). The paper investigates the application of reinforcement learning (RL) algorithms for QoE-aware flexible bandwidth allocation, which enables satellite service providers to minimize the allocated bandwidth while meeting the QoE requirements of their customers. By employing a time-varying queuing model, we formulated a stochastic optimization problem and applied Q-learning and state-action-reward-state-action (SARSA) reinforcement learning algorithms to determine the optimal bandwidth allocation. The findings indicate that while the algorithms exhibit similar convergence speeds, Q-learning slightly outperforms SARSA due to its more efficient bandwidth selection to meet the requirements. This demonstrates the potential of reinforcement learning as a valuable tool for optimal bandwidth allocation in satellite communications, thereby contributing to the ongoing improvement of service quality in this domain.**

***Index Terms*—Time-varying queuing, Flexible bandwidth allocation, Reinforcement learning**

I. Introduction

Next-generation satellite communication (SATCOM) networks are expected to meet high capacity, low latency, and seamless connectivity requirements for a diverse range of applications and services in remote and rural areas, as well as dense urban environments [1]. Given the increasing diversity of services and applications from IoT/cellular devices connecting to 5G/B5G satellite networks in very-near future, there is a critical need for flexible bandwidth allocation mechanism which can be capable of acknowledging application-specific QoE requirements to satisfy end-users' demands [2]. Such a QoE-aware mechanism can enhance network efficiency and utilization. This can be obtained by customizing network resources to specific services/applications, allowing SPs to adapt the assigned bandwidth to respond to time-varying user demands.

For satellite SPs aiming to reduce their costs and boost their revenue, it is essential to meet the QoE requirements of their users while minimizing bandwidth usage. To do so, satellite SPs must navigate several challenges encompassing bandwidth planning, and allocation optimization tasks, which are made complex by the irregular and unpredictable nature of time-varying data traffic generated by a variety of applications and services. Inspired by these complex challenges, this paper delves into the use of reinforcement learning by satellite service providers to effectively achieve QoE-aware flexible bandwidth allocation.

A few research studies in the literature are conducted on bandwidth minimization to meet specified Quality-of-Service (QoS) and QoE demands. Particularly, authors in [3] have delved into QoS demand-aware bandwidth minimization. However, their work does not take into account the time-varying demand and QoE requirements. Our recent works in [4], [5] have applied optimization techniques for QoE-aware cost-minimizing dynamic bandwidth allocation. Nevertheless, traditional optimization methods often fall short when it comes to leveraging past experiences, making accurate predictions, and handling extensive data for more adaptive and flexible network management [6], [7]. In another work, the authors in [6] applied multi-agent reinforcement learning for optimal use of bandwidth, power, and beam width in SATCOM systems. Similarly, the authors in [7] applied multi-agent reinforcement learning for dynamic bandwidth allocation for beam-hopping-enabled SATCOM systems. Yet, these studies do not account for QoE requirements, and they assumed average traffic demand, overlooking the fact that real traffic demands are time-varying.

To the best of our knowledge, a reinforcement learning-based framework for flexible bandwidth allocation that efficiently reduces the allocated bandwidth while satisfying the QoE requirements has not been investigated yet. This paper addresses these challenges and fills this gap in the literature. In particular, our work endeavors to apply Q-learning and SARSA reinforcement learning algorithms to tackle the difficult problem of QoE-aware flexible bandwidth allocation. By modeling the time-varying traffic arrivals and allocated bandwidth as a queueing model, we recast these design challenges into a stochastic optimization problem. The solution is subsequently provided using reinforcement learning, with careful deliberation

of the learning parameters, state space, action space, and reward functions. Lastly, we conduct numerical studies to validate the reinforcement learning model and illustrate the efficiency of our proposed frameworks.

The rest of the paper is organized as follows. In Section II, the system model and problem formulation are described. The flexible bandwidth allocation frameworks using reinforcement learning are proposed in Section III. In Section IV, the numerical results are discussed. Finally, Section V concludes the paper.

II. System Model and Problem Formulation

Let us consider a scenario where a satellite operator[1] intends to lease a flexible bandwidth with a maximum capacity of $W^{max}(bps)$ to an SP. This SP aims to offer a specific service to its users within the satellite beam's coverage area. We consolidate the demand from all application users seeking access to the satellite system into a flow of packets. The network operates in a time-slotted manner, with each time slot having a length of T_p (s) across an observation period of T. Constrained by processing capabilities, the satellite operator seeks to maintain a constant value of $W(t)$ for each cycle, which lasts M time slots. Consequently, we can establish $K = T/(M * T_p)$ bandwidth-unchanged cycles. Therefore, we can refer to $W(t)$ as W_k.

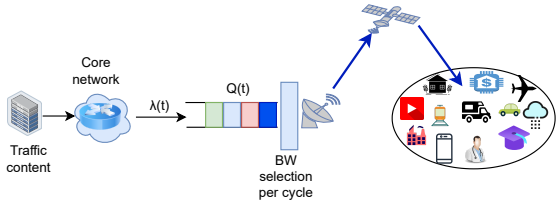

Fig. 1: Bandwidth allocation for a queued flow of packets.

A. Traffic and Queue Model

In our model, we consider the traffic demand from all service users within the coverage cell of a satellite as a stream of packets queuing to gain access to a SATCOM system, as illustrated in Fig. 1. We further assume this flow to have a time-varying arrival rate denoted as $\boldsymbol{\lambda}(t)$, i.e., $\lambda(t)$ is the number of packets of length L bits and $\lambda(t)$ changes over time. The data that arrives will be temporarily stored in the system buffer before being dispatched to the users via the SATCOM system's air interface. Given that the queue model is expressed in terms of the arrival rate and the stored data is transmitted to users in each time slot as packets of length L bits, we can define a relationship between the satellite bandwidth and the service rate as follows:

$$\mu(t) = W_k T_p / L. \tag{1}$$

Hereafter, we refer to the temporary number of packets present in the system buffer at time t as the queue length, $Q(t)$. With the assumption that the system buffer is devoid of packets at $t = 0$, and acknowledging that $Q(t)$ cannot drop below zero, we recognize that the dynamics of packet arrival in the flow exhibit variability [8],

$$Q(t+1) = min(max[Q(t) + \lambda(t) - \mu(t), 0], Q_{max} + 1), \tag{2}$$

where Q_{max} is the maximum queue length. We define the queue length for packets arriving after the buffer system is fully occupied as $Q_{max} + 1$.

B. QoE Requirement

In IoT-era wireless communication networks, where the majority of traffic flows come from services/devices requiring low latency, timely and emotionally relevant QoE reporting is challenging. Consequently, the traditional Mean Opinion Score (MOS)-based QoE reporting is no longer applicable as the end users are not humans. Instead, service providers measure QoE as the probability to which a device's target operating value range is fulfilled [9]. In this context, it can be assumed that all devices seeking to access the satellite require to tolerate a queue length of $\bar{Q}$ or less. Further, it is assumed that users (devices) require queuing delay requirements be violated with a probability less than $\bar{p}$. These requirements can be translated into the following conditions:

$$p(Q(t) \geq \bar{Q}) \leq \bar{p}. \tag{3}$$

where $\bar{p}$ is the probability of QoE requirement violation. In addition to considering the required threshold $\bar{Q}$ for maintaining good QoE, we should take into account the maximum buffer size, also known as Q_{max}. If the queue length exceeds this value, the system cannot function adequately, leading to the blocking and subsequent dropping of all newly arriving packets. In order to maintain the SATCOM's proper functioning, the probability of blocking a packet, also known as packet dropping probability, should not surpass $\bar{p}_d$. The dropping probability requirement due to an intolerable queue length can be defined as

$$p(Q(t) \geq Q_{max}) \leq \bar{p}_d. \tag{4}$$

where $\bar{p}_d$ is the target dropping probability. Regarding both QoE and system performance requirements, the design problem can be formulated as,

$$\min_{\mathbf{W}_k} \quad \sum_{\forall k} W_k \tag{5a}$$

$$\text{s.t.} \quad p(Q(t) \geq \bar{Q}) \leq \bar{p}, \forall k \tag{5b}$$

$$p(Q(t) \geq Q_{max}) \leq \bar{p}_d, \forall k. \tag{5c}$$

The formulated stochastic optimization problem poses challenges in terms of directly characterizing its convexity and finding a solution. As a result, it becomes necessary to simplify the constraints by expressing them in a more manageable form. Given that the queue length for the flow can be evaluated at every time slot, (3) can be interpreted as a probability of occurrence, given as follows:

$$\frac{n(Q(t) \geq \bar{Q})}{M} \leq \bar{p}, \forall k \tag{6}$$

[1]It is important to note that the results can be applied to other wireless communication systems as well.

where $n(.)$ indicates the number of occurrences in which the expression within the brackets is true. Similarly, the expression in (4) can be transferred into:

$$\frac{n(Q(t) \geq Q_{max})}{M} \leq \bar{p}_d, \forall k \tag{7}$$

III. Proposed Reinforcement Learning-based Flexible Bandwidth Allocation Frameworks

In this section, we elucidate the RL techniques, with an emphasis on their potential for effective use in the realm of flexible bandwidth allocation. This includes a detailed description of how they can help in satisfying QoE requirements while ensuring efficient use of bandwidth resources.

A. Q-learning Algorithm

Q-learning is one of the most important RL algorithms for tackling sequential decision-making problems. It is an off-policy and value-based algorithm that learns to make optimal decisions by interacting with the environment to estimate the values of different actions under different states [10]. In Q-learning, a learning agent interacts with an environment over a sequence of time steps. At each time step t, the agent observes the current environment state $\mathsf{s}(t)$ to select an action $\mathsf{a}(t)$. It then receives a respective reward of $\mathsf{r}(t)$ from the environment after taking the action $\mathsf{a}(t)$ and moves to a new state $\mathsf{s}(t+1)$. The objective of the agent is to maximize the long-term cumulative reward it receives over time. To apply Q-learning to our considered system, we define its key elements as follows.

1) Agent: The agent is the decision-making component. In our context, it is responsible for selecting the appropriate bandwidth level to fulfill the given requirements.

2) State space: The state space $\mathcal{S}$ represents the set of all possible states the agent may encounter. In our scenario, the queue length of the flow can assume values $q \in 0, 1, 2, ..., Q_{max}$, thus yielding an average per cycle queue length of Q_{mean}. Given that our bandwidth allocation strategy operates on a per-cycle basis, it would not be logical to define the state space in terms of the queue length per time slot. Instead, we aim to represent the state space based on the value of Q_{mean} and two required thresholds $\bar{Q}$, Q_{max}. In particular, by observing the average queue length values, we can construct a state space comprising 3 different states, as illustrated in Table I.

TABLE I: Q-learning state space description

State-space	description
1	$Q_{mean} \leq \bar{Q}$
2	$\bar{Q} < Q_{mean} \leq Q_{max}$
3	$Q_{mean} \geq Q_{max}$

3) Action space: The action space refers to all actions that the agent can take. The action of the agent is defined as the bandwidth selection for the communication process. Let A denote the action space which can be defined as $\mathsf{A} = \{0, W_1, W_2, ..., W_N\}$, where N is the number of available bandwidth levels. At each cycle k, the agent selects one of the bandwidth levels in A for transmission. To do this, it can use the ϵ-greedy strategy. In particular, considering a small number $0 < \epsilon < 1$, the agent explores (selects a random bandwidth from the action space) for a probability of ϵ, and it exploits (selects the bandwidth with the highest Q-value) for a probability of $1 - \epsilon$.

$$\mathsf{a} = \begin{cases} \text{random action} & \text{for probability } \epsilon, \\ \arg\max_{\mathsf{a} \in \mathsf{A}} \{Q(\mathsf{s}_k, \mathsf{a}_k)\}, & \text{for probability } 1-\epsilon, \end{cases} \tag{8}$$

where $Q(\mathsf{s}_k, \mathsf{a}_k)$ is the Q-value of state-action pair $(\mathsf{s}_k, \mathsf{a}_k)$ at cycle k. This probability decreases with the increase in the learning time at each episode by a factor of the epsilon decay (σ).

4) Reward: The rewards an agent receives indicate which bandwidth values are more valuable, and which bandwidth should be selected when the same state is visited in the future [10]. The allocated bandwidth by the agent changes its value at every cycle and the agent receives a corresponding reward for every selected bandwidth. Accordingly, the reward given to the RL agent for selecting a bandwidth of W_k per cycle can be given as:

$$\mathsf{r}_k = \begin{cases} Q_{mean} + 1/W_k, & \text{if } \frac{n(Q(t) \geq \bar{Q})}{M} \leq \bar{p}, \frac{n(Q(t) \geq Q_{max})}{M} \leq \bar{p}_d \\ 0, & \text{if } \frac{n(Q(t) \geq \bar{Q})}{M} \leq \bar{p}, \frac{n(Q(t) \geq Q_{max})}{M} > \bar{p}_d \\ 0, & \text{if } \frac{n(Q(t) \geq \bar{Q})}{M} \geq \bar{p}, \frac{n(Q(t) \geq Q_{max})}{M} < \bar{p}_d \\ -1, & \text{otherwise.} \end{cases} \tag{9}$$

The fact that Q_{mean} depends on the arrival rate and the allocated bandwidth, and our objective is to minimize the allocated bandwidth, we design our target reward function as the sum of Q_{mean} and the inverse of allocated bandwidth when the requirements are satisfied. We provide a value of zero if either QoE or packet drop probability requirements are violated. The agent is not expected to take an action that violates the QoE and packet drop probability requirements. For that reason, a small reward of -1 is given to the agent as compared to the case where one or both requirements are satisfied.

5) Q-table: The algorithm contains a Q-table as the main component to store the value of each state-action pair. The Q-table is first initialized to 0. As the same action taken at different cycles may result in different rewards, we need K Q-tables of size $S \times N$. In a certain state, every time the agent selects bandwidth (current action) is called an episode. After an episode, the agent compares the reward it got in this episode with that of the previous episode. Thus, it learns which bandwidth (target action) it must allocate in order to get the maximum reward. Accordingly, the Q-table of each cycle is updated as:

$$Q(\mathsf{s}_k, \mathsf{a}_k) = (1-\alpha) Q(\mathsf{s}_k, \mathsf{a}_k) + \alpha [R_k + \gamma max(Q(\mathsf{s'}_k, \mathsf{a}_k))], \tag{10}$$

where α is the learning rate, γ is the discount factor, R_k is the reward for the current action, $Q(\mathsf{s}_k, \mathsf{a}_k)$ is the estimated Q-value of the current state-action, and $max(Q(\mathsf{s'}_k, \mathsf{a}_k))$ is the best estimated Q-value of the next state-action. The estimated Q-values of the current and next state-action pairs update formula are estimates which are not very accurate at first. However, the reward received is concrete data that allows

Algorithm 1 Q-LEARNING ALGORITHM

Initial values:
- Set the values of $\bar{Q}$, Q_{max}, $\bar{p}$, $\bar{p}_d$.
- Define the state space and action space.
- Initialize a Q-table $Q(\mathsf{s}_k, \mathsf{a}_k)$ for all cycles.
- Initialize a learning rate (α), discount factor (γ) and a small number (ϵ)

for each episode **do**
  Set the initial state.
  for each cycle **do**
    Choose an action from the action space using the epsilon greedy method.
    $\mathsf{a}_k \longleftarrow argmax(Q(\mathsf{s'}_k, \mathsf{a}_k))$ for a probability of ϵ and a random action for a probability of $1-\epsilon$.
    Get the reward (R_k) and next state ($\mathsf{s'}_k$).
    Update the queue table as, $Q(\mathsf{s}_k, \mathsf{a}_k)$
    $= (1-\alpha)Q(\mathsf{s}_k, \mathsf{a}_k) + \alpha(R_k + \gamma max(Q(\mathsf{s'}_k, \mathsf{a}_k)))$.
    $\mathsf{s}_k \longleftarrow \mathsf{s'}_k$.
  end for
end for
Return Updated Q-table.

TABLE II: Considered parameter values.

Parameters	**values**
Learning rate	0.1 [11]
Discount factor	0.5 [11]
epsilon (ϵ)	1
epsilon decay (σ)	0.99
Number of episodes	1000
Number of cycles (K)	24
Target QoE violation probability ($\bar{p}$)	0.1
Target queue length ($\bar{Q}$)	40 packets
Maximum queue length (Q_{max})	60 packets
packet length (L)	300 bits [12]
Target packet drop probability ($\bar{p}_d$)	0.01 [13]
Maximum available bandwidth (W^{max})	500 Mbps
Number of bandwidth levels	25
Cycle duration (M)	1 Hour
Total observation time	1 day
Time slot duration (T_p)	1 Second

the agent to learn and improve its estimates based on actual experience with the environment. If we have an optimal Q-function, we get an optimal policy that shows the best action to take for each state in each flow.

B. State Action Reward State Action (SARSA) Algorithm

For comparison we consider the SARSA RL algorithm. SARSA is a value-based RL algorithm that uses a Q-table to store values for each state-action pair. With the stored values it helps us to train an agent indirectly by teaching it to identify which state-action pairs are more valuable [10]. Unlike Q-learning, which uses different policy for acting and updating the Q-table, SARSA uses the same policies for acting and updating the value function (updating policy) [10]. Accordingly, using SARSA the Q-table is updated as:

$$Q(\mathsf{s}_k, \mathsf{a}_k) = (1-\alpha)Q(\mathsf{s}_k, \mathsf{a}_k) + \alpha[R_k + \gamma(Q(\mathsf{s'}_k, \mathsf{a}_k))]. \quad (11)$$

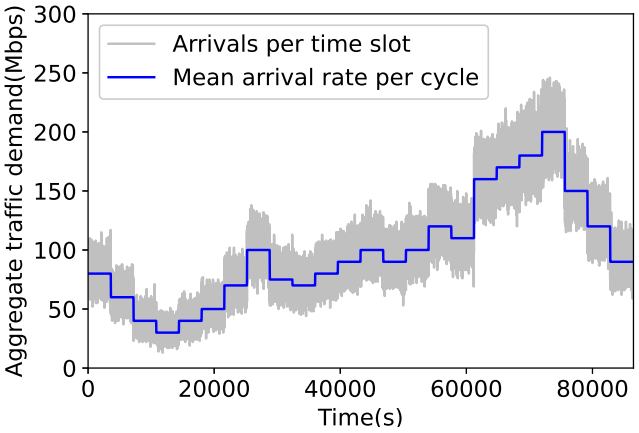

Fig. 2: Time-varying arrival rate per cycle and arrivals per time slot.

IV. SIMULATION PARAMETERS AND NUMERICAL RESULTS

This section provides the simulation and numerical results to analyze the performance of the considered RL techniques.

A. Simulation Setup and Parameters

In this section, we focus on the assembly of the necessary datasets for training the RL agent, as well as a discussion on the chosen parameter values. We collected traffic demand data featuring a time-varying mean, as demonstrated in Figure 2. This dataset aligns with the average per-hour traffic trend that was considered for various applications by the authors in [14]. From there, we generated a time-varying dataset, which includes a random arrival of packet flows, also featuring a time-varying mean. This dataset consists of $86,400$ traffic observations, covering a full 24-hour period and divided into one-hour cycles.

Inspired by [15], [16], we generated M random samples around the mean arrival rate for each cycle according to the Poisson process, as described in [8]. The inter-arrival time (IAT) of the considered Poison distribution is given by,

$$IAT = -log(1-\mathbf{R})/\lambda(k), \quad (12)$$

where $\mathbf{R}$ is a vector of random numbers between 0 and 1 in a cycle k and $\lambda(k)$ is the mean arrival rate at cycle k. The remaining considered parameter values are given in Table II.

B. Numerical Results and Discussion

In this subsection, we undertake an in-depth analysis to determine if the QoE and packet-dropping probability requirements are met within each cycle. This is conducted by implementing Algorithm 1 to obtain optimal bandwidth values and then calculating queue length values at every time t according to (2). Through this process, we aim to demonstrate the effectiveness and efficiency of our proposed reinforcement learning model, which has been designed and implemented specifically for the task of QoE-aware flexible bandwidth allocation in satellite SPs. Ultimately, this examination of real-time performance metrics will provide insight into the practical applicability of our approach and help guide further improvements.

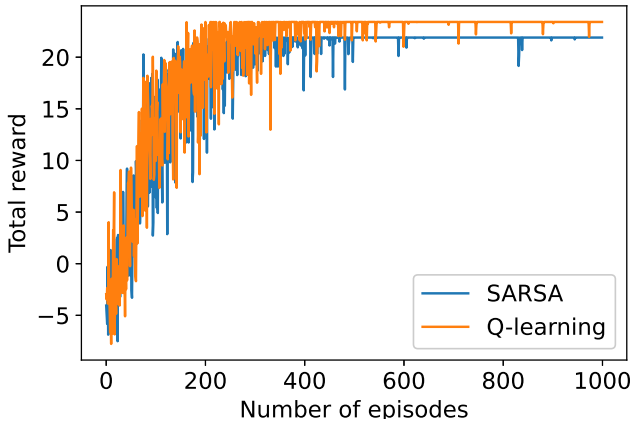

Fig. 3: Total reward versus the number of episodes.

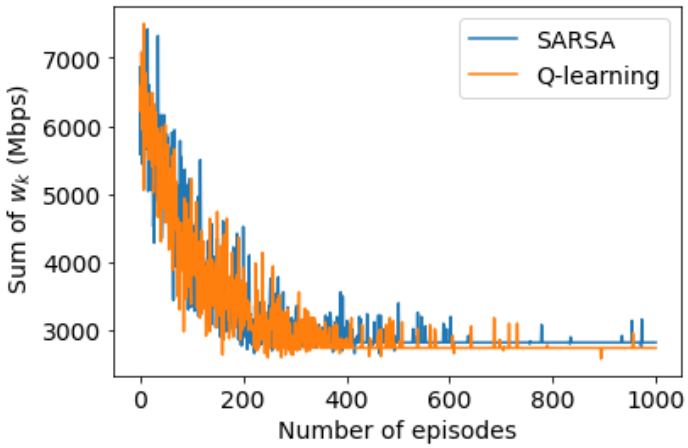

Fig. 4: Allocated bandwidth versus the number of episodes.

First, Fig. 3 illustrates the convergence plot as well as the total reward gained across all cycles ($\sum_{\forall k} r_k$) over 1000 episodes. This is observed while utilizing both Q-learning and SARSA algorithms. The results obtained reveal that both algorithms converge effectively. Based on the results Q-learning algorithm slightly outperforms SARSA as it yields more reward. In a similar vein, Fig. 4 demonstrates the convergence plot of both algorithms in terms of the total bandwidth allocated across all cycles ($\sum_{\forall k} W_k$) over the same 1000 episodes. From the plot, it can be discerned that the Q-learning algorithm slightly outperforms SARSA. This is because it manages to satisfy the stipulated requirements while utilizing less bandwidth.

Next, Fig. 5 displays the optimal bandwidth allocated using the Q-learning algorithm juxtaposed with the time-varying arrival rate. Upon analyzing the results, we find that the allocated bandwidth adjusts with each cycle in response to changes in traffic arrival rate, or demand. The difference between the mean arrival rate per cycle and the allocated bandwidth represents the amount of bandwidth required to meet the QoE and dropping probability requirements.

In addition, Fig. 6 illustrates the instantaneous and per-cycle queue length observed with the optimally allocated bandwidth obtained using Q-learning. The plot shows that the observed

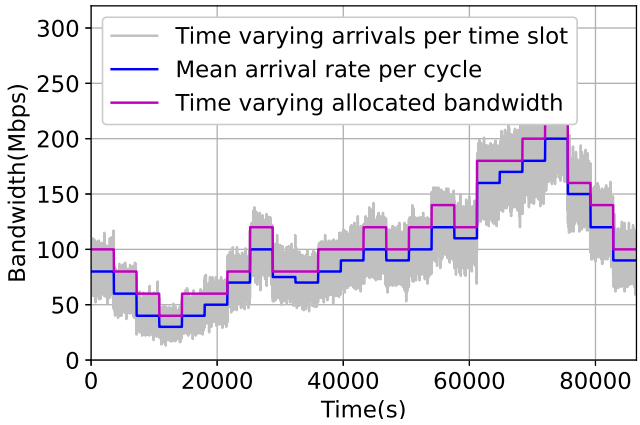

Fig. 5: Time-varying allocated capacity for time-varying demand using Q-learning.

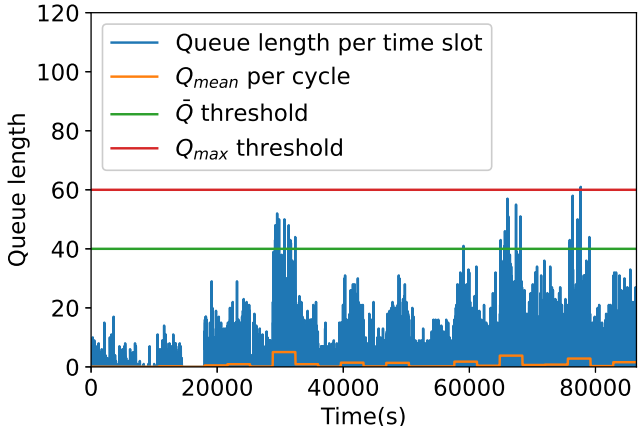

Fig. 6: Queue length per time slot and average queue length per cycle.

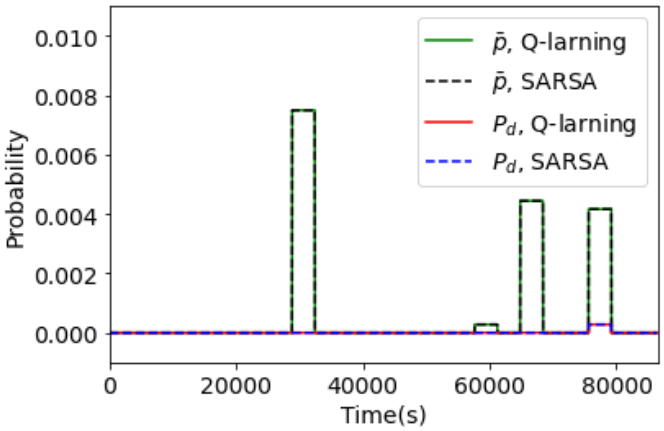

Fig. 7: Packet dropping probability and QoE requirements violation.

queue length, given the allocated bandwidth, successfully meets the $\bar{Q}$ and Q_{max} requirements. Similarly, Fig. 7 demonstrates the packet dropping and QoE requirement violation probability per cycle using both the Q-learning and SARSA algorithms. The results show that both algorithms meet the target $\bar{p}$ <= 0.1 and $\bar{p}_d$ <= 0.01 requirements, thereby proving their effi-

ciency. The subtle differences in performance between the two algorithms do not significantly impact the outcomes, further reinforcing the comparable efficiency of both approaches in this context.

Taken together, the close alignment between the observed queue lengths, QoE requirement violation probability, and packet-dropping probability requirements solidify the conclusion that both the Q-learning and SARSA algorithms are efficient and successful in meeting the QoE and dropping probability requirements.

V. Conclusion and Future work

This paper presents a novel RL-based approach for flexible bandwidth allocation in satellite service providers, aiming to minimize allocated bandwidth while addressing time-varying traffic flow with the critical packet dropping and QoE requirements. The QoE and packet dropping requirements, represented as a target queue length and maximum tolerable queue length that users are willing to wait for the service, are effectively managed using the proposed Q-learning RL algorithm. The obtained results demonstrate the superior performance of our Q-learning algorithm compared to the benchmark SARSA RL algorithm, as it satisfies the requirement with lower bandwidth consumption. The RL training process, implemented as offline learning with known arrival rates, considers all available data at once. Future work will extend the research to include multiple flows in multiple beams, enabling online learning scenarios. This advancement will contribute to further optimizing bandwidth allocation and meeting QoE requirements in satellite communication systems.

Acknowledgment

This work has been supported by the Luxembourg National Research Fund (FNR) under the project INSTRUCT (IPBG19/14016225/INSTRUCT).

References

[1] Nokia, "5G from space - The role of satellites in 5G," https://www.nokia.com/thought-leadership/articles/5g-space-satellites/, 2023, [Online; accessed 29-May-2023].

[2] B. Agarwal, M. A. Togou, M. Ruffini, and G.-M. Muntean, "Qoe-driven optimization in 5g o-ran enabled hetnets for enhanced video service quality," *IEEE Communications Magazine*, 2022.

[3] T. S. Abdu, S. Kisseleff, E. Lagunas, and S. Chatzinotas, "Power and bandwidth minimization for demand-aware geo satellite systems," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[4] T. M. Kebedew, V. N. Ha, E. Lagunas, J. Grotz, and S. Chatzinotas, "Qoe-oriented resource allocation design coping with time-varying demands in wireless communication networks," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 2022, pp. 1–5.

[5] T. M. Kebedew, "QoE-Aware Cost-Minimizing Bandwidth Renting for Satellite-as-a-Service enabled Multiple-Beam SATCOM Systems," https://www.techrxiv.org/articles/preprint/QoE-Aware_Cost-Minimizing_Bandwidth_Renting_for_Satellite-as-a-Service_enabled_Multiple-Beam_SATCOM_Systems/, 2023, [Online;].

[6] F. G. Ortiz-Gomez, D. Tarchi, R. Martínez, A. Vanelli-Coralli, M. A. Salas-Natera, and S. Landeros-Ayala, "Cooperative multi-agent deep reinforcement learning for resource management in full flexible vhts systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 335–349, 2022.

[7] Z. Lin, Z. Ni, L. Kuang, C. Jiang, and Z. Huang, "Dynamic beam pattern and bandwidth allocation based on multi-agent deep reinforcement learning for beam hopping satellite systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3917–3930, 2022.

[8] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent traffic steering in beyond 5g open ran based on lstm traffic prediction," *IEEE Transactions on Wireless Communications*, 2023.

[9] Rohde-Schwarz, "5G Quality of Experience," https://the-mobile-network.com/wp-content/uploads/2019/03/Rohde-Schwarz.pdf, 2022, [Online; accessed 23-April-2023].

[10] S. Dobilas, "Reinforcement Learning with SARSA — A Good Alternative to Q-Learning Algorithm," https://towardsdatascience.com/reinforcement-learning-with-sarsa-a-good-alternative-to-q-learning-algorithm, 2023, [Online; accessed 6-June-2023].

[11] D. Tran, S. K. Sharma, S. Chatzinotas, and I. Woungang, "Learning-based multiplexing of grant-based and grant-free heterogeneous services with short packets," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06.

[12] H. Jang, J. Kim, W. Yoo, and J.-M. Chung, "Urllc mode optimal resource allocation to support harq in 5g wireless networks," *IEEE Access*, vol. 8, pp. 126 797–126 804, 2020.

[13] M. Mozaffari, Y.-P. E. Wang, and K. Kittichokechai, "Blocking probability analysis for 5g new radio (nr) physical downlink control channel," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[14] J. B. Malone, A. Nevo, and J. W. Williams, "The tragedy of the last mile: Economic solutions to congestion in broadband networks," *IO: Empirical Studies of Firms & Markets eJournal*, 2016.

[15] B. Han, V. Sciancalepore, X. Costa-Perez, D. Feng, and H. D. Schotten, "Multiservice-based network slicing orchestration with impatient tenants," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 5010–5024, 2020.

[16] M. Alsenwi, E. Lagunas, and S. Chatzinotas, "Coexistence of embb and urllc in open radio access networks: A distributed learning framework," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 4601–4606.