QoE-Aware Flexible Capacity Allocation Planning for Multi-Service Satellite Communication Systems

Teweldebrhan Mezgebo Kebedew, *Student Member, IEEE*, Vu Nguyen Ha, *Senior Member, IEEE*,
Duc Dung Tran, *Member, IEEE*, Eva Lagunas, *Senior Member, IEEE*,
Joel Grotz, *Senior Member, IEEE*, Symeon Chatzinotas, *Fellow, IEEE*

***Abstract*—Satellite operators worldwide are in a race to deploy and enhance connectivity supporting diverse 5G applications and services, with success depending on the ability to deliver superior Quality of Experience (QoE) tailored to each service, despite limited network capacity. However, this effort is challenged by unpredictably fluctuating traffic demands, distinct packet arrival distributions across services, and evolving stochastic user QoE expectations. This paper addresses these challenges by formulating a statistical optimization problem that minimizes allocated capacity (intending to accommodate more users) while satisfying specific QoE requirements, such as queuing delay. To achieve this, we leverage packet queuing analysis within the buffer system of the SatCom gateway's forward link. Given the complexity of solving the problem directly, we first approximate its constraints using probabilistic analysis. Then, we propose a multi-agent Double Deep Q-Network (DDQN) algorithm that enables a more accurate representation of queue-length states and facilitates better decision-making by the agents. The approach leverages episodic training to ensure agents are well-prepared and optimized through simulations before being deployed in a real-time environment. Extensive simulation campaigns validate the effectiveness of our method, demonstrating clear improvements over benchmark algorithms.**

***Index Terms*—Capacity Allocation, Stochastic QoE, DQN, DDQN, Multi-service, Queuing Analysis, Blocking Probability.**

I. Introduction

Satellite communication (SatCom) networks have emerged as a promising technology, offering ubiquitous wireless connectivity in regions that lack terrestrial radio access service. They play an important role in improving resilience for different applications and services, significantly contributing to the advancement of future wireless communication networks. SatCom operators around the world are competing to provide the fastest, most reliable, and extensive 5G coverage, catering to heterogeneous services. Their primary goal is to satisfy users with diverse Quality of Experience (QoE) requirements under limited network resources [2].

In wireless communication systems, QoE-aware dynamic capacity allocation (CA) strategies have been considered advanced technologies that significantly impact network operators' revenue and rental costs incurred by service providers [3]. A key component in this process is the medium access control (MAC) layer, which manages packet transmissions and buffer queue status [4]. The MAC layer formats data for physical transmission and coordinates closely with the physical and radio link control (RLC) layers [5], facilitating effective scheduling, resource allocation, and traffic prioritization [6], [7]. By flexibly prioritizing capacity based on service-level agreements (SLAs) and user QoE, SatCom operators can implement proactive and effective CA policies. Such policy design is crucial, as allocating the minimum capacity needed to ensure satisfactory QoE effectively reduces capital and operational expenditures [8], [9]. The primary challenges are due to the following.

(i) Network operators often struggle with defining metrics to accurately model the QoE, a challenge exacerbated by evolving stochastic quality expectations [10]–[12].
(ii) Traffic demands fluctuate unpredictably over time, making it difficult to pre-allocate capacity without risking over-provisioning or service degradation.
(iii) Different services exhibit distinct packet inter-arrival characteristics, complicating the modeling of burstiness and increasing the complexity of traffic handling [13].
(iv) Dynamically coordinating limited capacity among multiple services is challenging due to competing demands, which complicate the prioritization of services during congestion, necessitating more intelligent and adaptive CA policies.

To handle these challenges, SatCom operators must model the time-varying characteristics of packet arrivals and inter-arrival times for various service types. Assuming simplistic or homogeneous traffic models, such as applying a Poisson distribution to all traffic, fails to capture the burstiness and diversity of real-world traffic [14], [15]. To address this, we tackle the more realistic and complex nature of SatCom traffic by incorporating diverse inter-arrival distributions and service-specific behaviors. Yet, due to the inherent uncertainty in user demand, real-time performance indicators are still necessary for effective decision-making. Due to limited capacity, satellites cannot admit every incoming packet instantly. Instead, packets are buffered, and traffic fluctuations lead to time-varying queue length (QL) within the system. In this context, QL naturally emerges as a key metric. The QL serves as a critical indicator

The preliminary results of this manuscript were presented in Globecom 2023 [1]. This work has been supported by the Luxembourg National Research Fund (FNR) under the project INSTRUCT (IPBG19/14016225/INSTRUCT).

T. M. Kebedew, V. N. Ha, E. Lagunas, D. D. Tran, and S. Chatzinotas are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 1855 Luxembourg Ville, Luxembourg (e-mail: teweldebrhan.kebedew@uni.lu; vu-nguyen.ha@uni.lu; eva.lagunas@uni.lu; duc.tran@uni.lu; Symeon.Chatzinotas@uni.lu).

J. Grotz is with SES, Chateau de Betzdorf, Betzdorf 6815, Luxembourg (e-mail: Joel.Grotz@ses.com).

of network congestion, providing insight into allocated capacity, queueing delay, packet drop probability, and overall QoE. This makes it a valuable indicator to optimize network performance [16], [17]. A high QL suggests overload and possible QoE degradation, while a low QL indicates resource underutilization, suggesting the potential for capacity reallocation.

Beyond modeling traffic patterns, modeling QoE itself presents additional complexity. While the Mean Opinion Score (MOS) has traditionally been used to measure perceived quality on a scale from 1 (bad) to 5 (excellent) [10], [12], its subjectivity, context sensitivity, and vulnerability to outliers limit its reliability in dynamic environments. Though still in the subjective estimation, extending QoE analysis from MOS to statistical indicators, such as the probability of "good or better" (GoB) and "poor or worse" (PoW) ratings, represents a meaningful advancement in capturing user satisfaction more quantitatively [18], [19]. To address these subjective issues, objective QoE metrics can offer a more reliable and automated approach to assessing user experience [20]. Objective metrics, such as latency and packet loss, provide quantifiable data that can be automatically monitored in real-time to guide capacity decisions. Despite these improvements, traffic demand and the available resources are inherently time-varying and uncertain, making it challenging to ensure that a pre-allocated capacity consistently satisfies the requirements of users. To address these limitations, a stochastic QoE evaluation framework that quantifies the probability of meeting objective target requirements can be employed [11], [12]. This approach enables a more realistic and comprehensive understanding of the relationship between unpredictable traffic patterns, CA, and QoE performance, facilitating more adaptive and effective capacity management in dynamic network environments.

Telecom operators engage in periodic planning and optimization of available capacity to ensure efficient utilization and meet SLAs. These optimization activities occur at varying intervals, ranging from minutes, hours, and days to monthly, guided by operator policies and network stability [21], [22]. Despite enhancing capacity utilization efficiency, these methods often exhibit a reactive nature, leading to potential resource underutilization or over-utilization between optimization intervals. Moreover, conventional optimization methods have limitations in harnessing historical data, accurately predicting outcomes, and managing large datasets for adaptable and flexible capacity management. These challenges highlight the growing need for adaptive, data-driven strategies that can dynamically respond to changing network conditions and improve the flexibility and efficiency of capacity planning [23].

Despite extensive optimization efforts, a significant advancement toward achieving adaptive and proactive CA is the exploitation of Machine Learning (ML) techniques, particularly Reinforcement Learning (RL) algorithms. Advanced Deep Reinforcement Learning (DRL) algorithms such as Deep Q-Network (DQN), Double DQN (DDQN), Dueling DDQN (3DQN), Proximal Policy Optimization (PPO), and actor-critic models have shown promise in this domain [24]–[29]. Among these, DQN variants, particularly DDQN and 3DQN, are well suited to scenarios with high uncertainty and stochastic QoE requirements [30]. Using these advanced techniques, we can quickly obtain near-optimal CA solutions, even amidst significant fluctuations in network traffic demands. This is achieved by preemptively learning the intricate relationship between traffic patterns of different services and optimal CA, ultimately enhancing the QoE for users across various services.

To enable centralized management with decentralized decision-making, where individual agents independently optimize CA for different services, and for additional reasons explained in Section III, this paper proposes leveraging a multi-agent DDQN that integrates real-time QL monitoring with service-specific modeling of packet inter-arrival distributions and stochastic QoE metrics. The proposed approach aims to proactively optimize CA decisions, thus improving QoE for diverse service requirements and overall network efficiency.

A. Related works

Dynamic resource allocation has been extensively studied in the literature and remains an evolving area of research aimed at addressing emerging challenges.

1) Non-QoE-Aware Dynamic Resource Allocation: Most works on resource allocation primarily concentrate on augmenting available capacity to meet aggregated traffic demand by improving the utilization efficiency of power, bandwidth, beam direction, or joint management. They aim to minimize the mismatch between offered traffic and required capacity. Techniques, such as beam hopping [17], [31], beam illumination [32], beam scheduling [33], joint user scheduling, power allocation and precoding [34], and joint beam, power, and bandwidth allocation [35], are employed. Although these works significantly contribute to dynamic resource allocation and resource utilization efficiency, they neglect an important aspect of resource allocation: QoS and QoE.

2) QoE-Aware Dynamic Resource Allocation: Resource allocation to meet QoE demands has been extensively studied in the literature. For instance, the authors in [39] explored the trade-off between transmission rate and service price using a fuzzy-based approach, demonstrating how fuzzy logic can optimize resource allocation to balance cost and QoE. Similarly, the authors in [38] examined the trade-off between allocated bandwidth and the QoE requirements of ultra-high-definition (UHD) video services using game theory, providing insights into how game-theoretic models can be used to allocate bandwidth efficiently. In [40], the authors investigated QoE-aware pricing, power allocation, and admission control to ensure a minimum data rate to maintain the QoE of video call services, highlighting the importance of integrating pricing strategies with resource allocation to enhance user satisfaction. Furthermore, QoE-driven resource allocation frameworks for video streaming on 5G networks are proposed in [36], [37]. Although these methods can improve QoE and resource utilization efficiency, they primarily consider aggregate traffic demand rather than demand per service. Additionally, these works do not account for multiple services, and the inter-arrival distribution of traffic demand is not discussed.

3) QOE-Aware Dynamic Resource Allocation for Multiple services: The complexity and challenge of simultaneously fulfilling the QoE requirements of multiple co-existing services

TABLE I: Comparison of Schemes

Related Work	Multiple Services	QoE Consideration			Service Priority	Varying Arrival Distribution
		MOS	Stochastic	Service Specific		
[17], [31]–[35]						
[36]–[40]		✓				
[41]	✓					
[42]	✓				✓	
[43], [44]	✓	✓				
[45], [46]	✓	✓			✓	
[47]	✓		✓			
This Work	✓		✓	✓	✓	✓

is greater than that of fulfilling the requirements for a single service or aggregate QoE. The work in [41] presents a dynamic channel reservation approach grounded in a DQN tailored for multi-service low earth orbit (LEO) satellite communication systems. Authors in [42] also worked on resource allocation for multiple services with priority. Another work [43] focuses on QoE-aware resource allocation for multiple IoT services. However, these works evaluate QoE using a cumulative score, which overlooks the distinct QoE requirements of each service. The authors in [44] worked on a more detailed categorization of QoS provisioning by incorporating radio resource, storage, and computing resource allocation for multiple coexisting services. Although their work highlights the importance of resource allocation for maintaining service-specific QoS, it assumes a homogeneous traffic model based on a Poisson distribution for all services and treats the transmission rate as constant, represented by a fixed mean packet arrival rate. This assumption overlooks the diverse arrival distributions and the time-varying nature of packet transmission.

Additionally, the works in [45], [46] introduce a self-tuning algorithm for optimizing QoE across multiple services in LTE networks by adjusting service priority parameters. However, these works estimate QoE solely based on the basic MOS approach and they neglect the consideration of packet inter-arrival distribution. In [47], we exploit optimization techniques to develop a novel dynamic QoE-aware CA algorithm addressing time-varying traffic in multi-beam multi-service SatCom systems to minimize the operation costs and satisfying QoE demands. However, the work neglects the difference in arrival distribution among flows and assumes that all flows adhere to the Poisson process. In addition, the study assumes the *first-come-first-served* queuing analysis approach which contrasts with our current prioritized queuing analysis. Table I provides a summary of related works, highlighting the limitations of existing literature and the key contributions of our approach.

To the best of our knowledge, the existing literature has not yet explored QoE-aware flexible CA planning that accommodates multiple coexisting services, each with distinct QoE requirements and varying packet inter-arrival distributions. The method proposed in our work aims to fill this gap.

B. Contributions

In this study, we focus on QoE-aware flexible CA planning, leveraging time-varying QL dynamics and DDQN to derive an optimal CA planning policy. Our primary contributions are:

- Diverse inter-arrival distribution scenarios: We model multiple co-existing services, each characterized by distinct packet inter-arrival time distributions. This nuanced approach accurately captures the traffic burstiness and variability inherent in real-world scenarios, providing a more comprehensive understanding of peak and off-peak traffic arrivals.
- Customized stochastic QoE requirements: We introduce a novel dimension by considering the diverse, uncertain QoE requirements associated with different services. Recognizing the varying sensitivities of services to allocated capacity enables us to tailor our CA planning strategy and to optimize user experience based on specific preferences.
- Service prioritization: Developed a novel CA planning mechanism using DDQN, integrating QoE-awareness and priority-based service provisioning to efficiently manage capacity across diverse services, mitigating performance degradation during congestion. During high-congestion periods, priority levels can be dynamically adjusted to ensure essential services receive adequate resources.
- Prediction followed by optimization: The traffic demand expected in future time slots is not known in advance, making it difficult to apply optimization techniques for CA online. Therefore, in this work, we employed a multi-horizon LSTM model for traffic prediction and adopted the Lagrangian duality optimization approach in [47] to incorporate customized QoE requirements for the services. This method was used as a benchmark to compare against our proposed approach.

In summary, our proposed approach leverages multi-agent DDQN for adaptable capacity dimensioning to aid satellite service providers in optimizing capacity for delivering multiple services while prioritizing QoE requirements. Our preliminary studies of this objective have been presented in [1], which utilized the conventional Q-learning approach for single-flow single-beam SatCom systems. In this current work, we aim to expand upon those findings by considering multibeam, multi-service schemes that require larger state- and action-space ML models, which exceed the capabilities of the Q-learning approach. We introduce double DQNs and provide robust demonstrations along with additional benchmark comparisons.

The remaining sections of the paper are organized as follows. Section II presents the system model and problem formulation. Section III delves into the queuing analysis, problem approximation, and DRL methods. Subsequently,

TABLE II: LIST OF KEY NOTATIONS

Notation	Definition
$a^b_{f,k}$	Current action by agent for flow f of beam b at cycle k.
B	Total number of virtual beams.
F	Number of data flows per beam.
K	Number of cycles.
L_f	Packet length of packets in flow f.
M	Number of TSs per cycle K .
$n^b_{f,\text{Blk}}$	Number of flow f of beam b packets blocked.
$n^b_{f,\text{QoE}}$	Number of flow f of beam b packets with QoE violated.
$\bar{P}_{\text{Blk}}$	Target blocking probability.
$\bar{P}_{\text{QoE}}$	Target QoE violation probability.
$\bar{q}^b_f$	Mean QL per cycle for flow f of beam b.
q_{max}	Maximum buffer length.
q^f_{QoE}	Target QL of packets in flow f.
$q^b(t)$	QL in beam b at time t.
$r^b_{f,k}$	Local reward of beam b for flow f at cycle k.
$R^b_{f,k}$	Total reward of beam b for flow f at cycle k.
$s^b_{f,k}$	Current state of beam b for flow f.
$s'^b_{f,k}$	Next state of beam b for flow f.
T	Total observation time (seconds)
T_{ts}	TS duration (seconds).
W^{max}	Maximum available capacity of the satellite (bps).
W^{total}_b	Total capacity per beam (bps).
$W(t)$	Total allocated capacity at time t.
$W^b_f(t)$	Capacity allocated for beam b due to flow f at time t.
$W^b_{f,k}$	Capacity allocated for beam b due to flow f at cycle k.
$\lambda^b_f(t)$	Arrival rate of data packets due to flow f of beam b.
$\lambda^b(t)$	Arrival rate of data packets to beam b.
$\mu^b_f(t)$	Service rate at beam b for flow f.
$\varphi^b_{f,k}$	QoE violations due to flow f of beam b at cycle k.
$\Theta^b_{f,k}$	BP violations due to flow f of beam b at cycle k.
$\Lambda^b(t)$	Total arrival rate to beam b.
ω, $\hat{\omega}$	The parameters of the current and target DQN network.
θ, θ^-	The parameters of the current and target DDQN network.
α	Learning rate.
γ	Discount factor.
β, β_1	Scale parameters of Pareto and Weibull distribution.
η, η_1	Shape parameters of Pareto and Weibull distribution.
τ	Inter-arrival time.

Section IV provides the numerical results. Lastly, Section V concludes the work. For ease of reference, a list of key notations used in this paper is provided in Table II.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a Geostationary Earth Orbit (GEO) satellite capable of providing multiple radio access services over B radiation beams to users randomly distributed across these beams' coverage areas. In this scheme, the downlink traffic demand (transmission requests) from various users within the coverage of beam b comes from the core network and is aggregated at the top of Layer 2, within the Service Data Adaptation Protocol (SDAP) [6] and represented as multiple packet flows. We assume that there are F different data flows, each with different packet lengths, arriving at the gateway buffer, specific to a particular satellite beam. These data flows are assumed to originate from user requests within the coverage area of that beam. The gateway is aware of the users residing in each beam and accordingly routes packets from external networks to the appropriate buffers. In this system, the dynamic CA mechanism is developed centrally at the network controller located on the ground, e.g., integrated at the gateway side as shown in Fig. 1. This centralization enables informed decision-making in real time based on a holistic view of network conditions, ensuring efficient resource allocation and adaptability to dynamic traffic flows. Using this mechanism, the SatCom operator can dynamically and efficiently allocate varying amounts of capacity to different beams based on the needs of the flows within them, satisfying user QoE requirements[1] over time and minimizing operator costs.

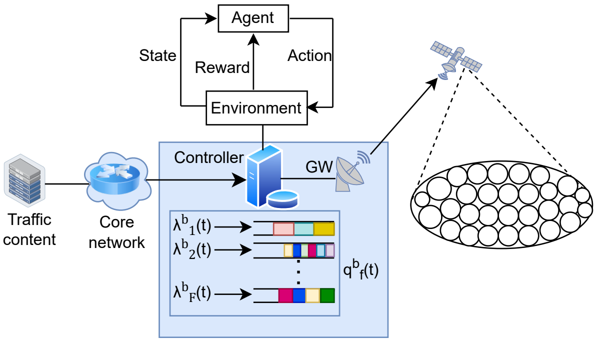


Fig. 1: CA for a queued flow of packets using RL.

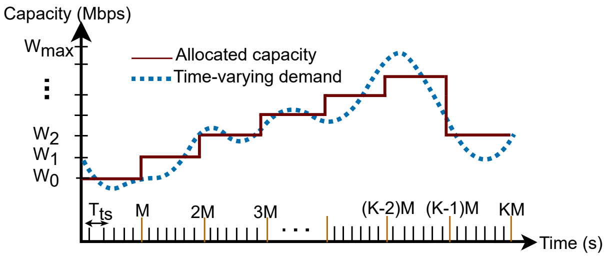


Fig. 2: The cycle-based capacity allocation framework.

A. Time-Varying Capacity Allocation

Let W^{max} (bps) denote the maximum SatCom capacity that the operator can allocate to serve F traffic flows in all beams. Each flow in each beam is assigned a portion of the capacity at any time t, denoted by $W^b_f(t)$, which can range from 0 to W^{max}, i.e., $0 \leq W^b_f(t) \leq W^{\text{max}}$. In addition, it is imperative to allocate capacity to each type of service (flow), ensuring that the total capacity assigned to all flows across all beams does not exceed the maximum available capacity of the operator, which yields the following constraint.

$$\sum_{b=1}^{b=B} \sum_{f=1}^{f=F} W^b_f(t) \leq W^{\text{max}}. \tag{1}$$

As described in Fig. 2, the network operates for a period of T seconds, divided into multiple time-slots (TS), termed the transmission time. Herein, each TS lasts for a duration of T_{ts} seconds. For various technical, operational, and economic reasons, limitations of the satellite system impose a minimum granularity of time to reconfigure onboard resources. Therefore, we assume that $W^b_f(t)$ for all (b, f) remains constant for the

[1] The user QoE requirement will be explained later in the following section.

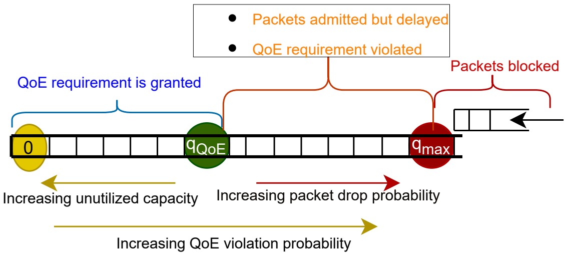

Packets admitted but delayed
QoE requirement violated
Packets blocked
QoE requirement is granted
0
q_{QoE}
q_{max}
Increasing unutilized capacity
Increasing packet drop probability
Increasing QoE violation probability

Fig. 3: Impact of QL on QoE and blocking probability.

duration of a cycle of M TSs and it can only be reset at TS indices $t_r \in \{0, M, 2M, ..., kM, ...\}$. This can be written as

$$W_f^b(t) = W_f^b(kM) \text{ if } t \in \big((k-1)MT_{\text{ts}}, kMT_{\text{ts}}\big], \qquad (2)$$

where $k = 1, ..., K$ and $K = T/(MT_{\text{ts}})$. In the following, the paper refers to $W_f^b(kM)$ as $W_{f,k}^b$, representing the flow capacity f allocated to the beam b in cycle k. The design framework considers a period that spans K cycles during T seconds.

B. Queuing Model

This study addresses the challenge of managing heterogeneous traffic demand varying in time, originating from various services, each of which exhibits diverse patterns of arrival rate distribution and distinct QoE requirements. To address this issue, the demands from end users in different services are classified by service type and modeled as service-specific flows of packets queued to access each beam.

1) Arrival Model: The data flows are categorized according to their respective statistical parameters, including arrival rates, service rates, and packet lengths. Consider F data flows corresponding to F services that tend to access each of B beams, as shown in Fig. 1. We further assume that the flow f $(f = 1, ..., F)$ transports data packets of length L_f bits. Additionally, flow f corresponding to beam b has a time-varying arrival rate of $\lambda_f^b(t)$ packets.

2) Service Rate: The service rate in the number of packets for a flow f entering to beam b at any time t can be expressed as a function of the allocated capacity as follows:

$$\mu_f^b(t) = W_f^b(t)T_{\text{ts}}/L_f. \qquad (3)$$

3) Corresponding Queue Length: Consequently, the QL of packets due to flow f of beam b at any time t is given as

$$q_f^b(t+1) = \min\Big(\max\big(q_f^b(t) + \lambda_f^b(t) - \mu_f^b(t), 0\big), q_{\max}\Big), \quad (4)$$

where $\max\big(q_f^b(t) + \lambda_f^b(t) - \mu_f^b(t), 0\big)$ is to ensure that the QL is always non-negative. Similarly, the minimization process assumes that the QL will not exceed the maximum. Here, $q_{\max}$ indicates the maximum buffer length. Having $q_f^b(t)$, the mean QL of flow f in beam b over a cycle k can be simply calculated as

$$\bar{q}_{f,k}^b = \frac{\sum_{t=(k-1)M}^{t=kM} q_f^b(t)}{M}. \qquad (5)$$

C. QoE Requirements

As mentioned in Section I-A, state-of-the-art methods commonly use the MOS as a standard metric for QoE estimation. The basic MOS approach may not accurately capture QoE, as it relies on users' subjective expectations of fulfilling certain requirements [2], [12]. It is also costly, time-consuming, and impractical for large-scale environments due to the extensive number of participants required for the experiment [40]. QoE can also be evaluated by considering user's expectations and objective performance metrics like throughput, latency, and jitter. In SatCom networks, latency is a key QoE metric, especially for real-time services. Since total latency comprises queuing and propagation delays, and GEO systems have near-constant propagation delay, QL becomes the main and practical indicator of service latency.

As depicted in Fig. 3, consider a buffer with a specified size, a target QL q_{QoE} predefined based on SLA, and a maximum QL $q_{\max}$. At any time t, the traffic demand fluctuates, leading to a dynamic QL at the buffer. If we assume no previously queued packets are available at the buffer system, gradually as the QL grows from zero toward q_{QoE}, the queuing delay increases, progressively violating QoE requirements. When the QL extends from q_{QoE} towards $q_{\max}$, QoE requirements are increasingly violated, and the probability of packet blocking grows, signaling the need for additional CA. Once the queue exceeds $q_{\max}$, packets are blocked and lost, further degrading service quality. Hence, in our work, we define QoE as the probability that data packets of a specific service type will not encounter a QL exceeding a threshold q_{QoE}, upon their first arrival in the gateway buffer, along with a predetermined probability of violating it. The probability that the expected QL exceeds the required target should not surpass a designated threshold known as the probability of QoE violation. This requirement can be given as

$$\mathbf{Prob}\{q_f^b(t) \geq q_{\text{QoE}}^f\} \leq \bar{P}_{\text{QoE}}, \forall(t, b, f). \qquad (6)$$

where q_{QoE}^f and $\bar{P}_{\text{QoE}}$ stand for the target QL requirement of flow f (i.e. service f), and the threshold probability of QoE violation. Here, we assume that the different flows corresponding to different services have different QoE requirements (q_{QoE}^f). Similarly, the probability of the QL reaching or exceeding the maximum QL must be lower than another specified threshold of blocking probability. This can be expressed as

$$\mathbf{Prob}\{q_f^b(t) \geq q_{\max}\} \leq \bar{P}_{\text{Blk}} \quad , \forall(t, b, f). \qquad (7)$$

where $\bar{P}_{\text{Blk}}$ is the target blocking probability[2] of all flows.

D. Problem Formulation

To enhance the revenue of satellite operators by accommodating more users, we need to allocate the minimal capacity

[2] In practice, different services have different blocking probability requirements. However, in this context, we associate the blocking probability with the likelihood that the QL exceeds the maximum buffer size, which is assumed to be constant across all services

that still meets QoE and blocking probability requirements. Hence, the problem can be formulated as follows

$$\min_{\{W^b_{f,k}\}'s} \quad \sum_{\forall b}\sum_{\forall f}\sum_{\forall k} W^b_{f,k} \tag{8a}$$

$$\text{s.t.} \quad \textbf{Prob}\{q^b_f(t) \geq q^f_{\mathsf{QoE}}\} \leq \bar{P}_{\mathsf{QoE}}, \forall(k,b,f), \tag{8b}$$

$$\textbf{Prob}\{q^b_f(t) \geq q_{\max}\} \leq \bar{P}_{\mathsf{Blk}}, \forall(k,b,f), \tag{8c}$$

$$\text{Constraint (1)}, \tag{8d}$$

where constraint (8b) stipulates that the probability of the QL exceeding q^f_{QoE} throughout the period must not surpass $\bar{P}_{\mathsf{QoE}}$ for all flows across all beams and cycles. Likewise, (8c) ensures that the blocking probability of packets in each flow within beam b at every cycle and time t remains below the target. Constraint (8d) ensures that the total allocated capacity to all flows in all beams at every cycle k cannot exceed the available satellite spectrum capacity. The primary challenge in solving the problem arises from the stochastic nature of the formulas in constraints 1 and 2, rendering it difficult to provide explicit solutions. Therefore, to solve the problem, it is necessary to approximate the constraints with equivalent expressions.

A certain number n of newly arrived packets at time t is blocked if the total packets comprising accumulated packets from the previous TS and newly arriving packets, minus the processed packets (service rate), exceed the maximum QL. Hence, the number of flow f packets blocked from accessing beam b at time t, denoted as $n^b_{f,\mathsf{Blk}}(t)$, is given by

$$n^b_{f,\mathsf{Blk}}(t) = \max\left(0, q^b_f(t-1) + \lambda^b_f(t) - \mu^b_f(t) - q_{\max}\right). \tag{9}$$

where $q^b_f(t-1)$ is the QL in the previous TS for flow packets f tending to the access beam b. Similarly, the number of flow f of beam b packets with QoE requirements violated at time t, denoted as $n^b_{f,\mathsf{QoE}}(t)$, is given by

$$n^b_{f,\mathsf{QoE}}(t) = \max\left(0, q^b_f(t-1) + \lambda^b_f(t) - \mu^b_f(t) - q^f_{\mathsf{QoE}}\right). \tag{10}$$

For short periods, such as a single TS, $n^b_{f,\mathsf{QoE}}(t)$ and $n^b_{\mathsf{Blk}}(t)$ may not represent meaningful averages to calculate the probability of blocking and the probability of QoE violation. However, over a sufficiently large number of TSs, the problem constraints can be approximated by their probabilities of occurrence as

$$\varphi^b_{f,k} = \textbf{Num}\{q^b_f(t) \geq q^f_{\mathsf{QoE}}\}/M \leq \bar{P}_{\mathsf{QoE}}, \forall f,b,k, \tag{11}$$

and

$$\Theta^b_f(k) = \textbf{Num}\{q^b_f(t) = q_{\max}\}/M \leq \bar{P}_{\mathsf{Blk}}, \forall f,b,k, \tag{12}$$

where **Num**{.} indicates the number of occurrences the expression is true.

III. DRL for Optimal Capacity Allocation

This section explores how DRL, specifically DQN, DDQN, and 3DQN, can be applied to develop a flexible CA approach for establishing optimal policies in proactive capacity planning across multiple QoE-centric satellite services. RL is a ML approach that allows an agent to reach a specific objective by maximizing long-term rewards through trial-and-error interactions with its environment [26]. The agent interacts by choosing actions from its available action space based on its current state. Each action results in a corresponding reward and a transition to a new state. This process is repeated until the agent's learning process converges to an optimal policy, maximizing the average reward. We explore two different multi-agent models: (1) one agent per flow, resulting in a total of BF agents, and (2) one agent per beam, a total of B agents. The choice of the number of agents involves a trade-off: Assigning BF agents (one agent per flow) provide fine-grained control and smaller action spaces per agent but at the cost of significant computational complexity. In contrast, using B agents (one agent per beam) strikes a balance between granularity and complexity, offering moderate action spaces while potentially facing challenges in fair resource allocation among flows within each beam. While a single centralized agent could theoretically manage the entire system, this approach is not considered in our work due to the high training complexity associated with the large action space, as discussed in Section III-A4.

A. Elements of RL Model

*1) **Environment**:* The environment is the considered B-beam satellite system, as depicted in Section II, which imposes specific constraints on CA. The dynamics of the environment encompass the evolution of the queue state in response to allocated capacity and external factors, including packet arrival rates and traffic patterns. These environmental dynamics are critical for the RL model to adapt and optimize CA, thereby ensuring the target QoE requirements.

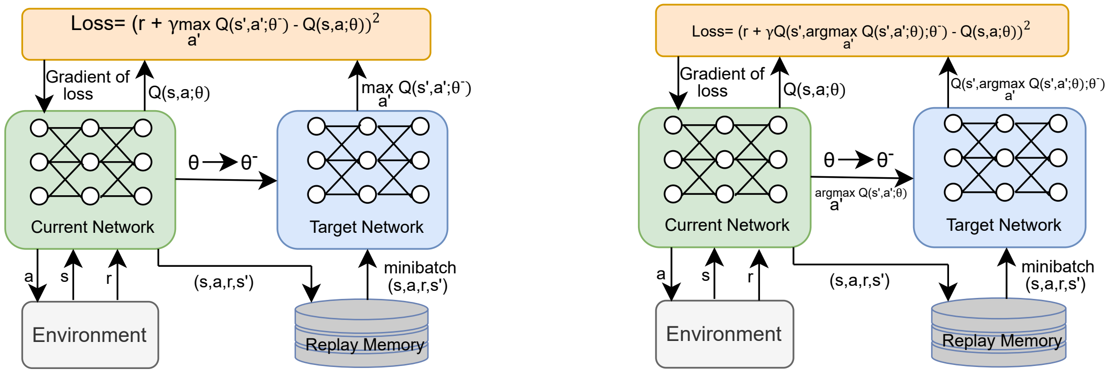
*2) **Agent**:* In our context, an agent (whether an agent of a single flow or an agent of all flows accessing a particular beam) refers to a CA manager and decision maker, co-located at the gateway. Its role is engaging with the environment to develop an optimal CA policy that minimizes capacity consumption while satisfying users' requirements on the QoE and blocking probability.

*3) **State**:* The state is defined as the specific instance of congestion in the satellite environment, which is determined by measuring the QL at the gateway. In particular, the QL of flow f of beam b at TS t ($q^b_f(t)$) can be calculated based on (4). However, the CA system is designed to work per cycle. Therefore, we model the state of an agent managing the flow f accessing beam b in cycle k, denoted as $s^b_{f,k}$ as the mean QL of all time intervals within a cycle k, that is, $s^b_{f,k} = \{\bar{q}^b_{f,k}\}$. Similarly, the state of an agent that manages the CA of F flows accessing beam b at cycle k is expressed as

$$s^b_k = \{\bar{q}^b_{1,k}, \bar{q}^b_{2,k}, \bar{q}^b_{3,k}, ..., \bar{q}^b_{F,k}\} \in \mathcal{S}, \ \forall(b,k), \tag{13}$$

where $\mathcal{S}$ denotes the set of all possible states of the agent. While the state captures the QL values for all flows for a particular cycle, the agent implicitly learns temporal patterns through accumulated experiences stored in the replay buffer as detailed in Section III-B. This enables the agent to account for time-varying traffic trends without requiring explicit historical information in the state representation.

*4) **Action**:* In our context, an action refers to selecting and allocating a specific capacity value from a given action space to satisfy the quality requirements of users. The action space



(a) DQN.

(b) DDQN.

Fig. 4: Training process in DQN and DDQN.

$\mathcal{A}$ is a set of discrete capacity levels from which agents can select to meet QoE requirements.

$$\mathcal{A} = \{W_1, W_2, \dots, W_N\}, \tag{14}$$

where N is the number of capacity levels. Thus, for the one agent per flow, the action of an agent for flow f of beam b at a cycle k, denoted $a^b_{f,k}$, can be given by

$$a^b_{f,k} \in \mathcal{A}. \tag{15}$$

Similarly, for the one agent per beam model, the action of an agent managing F flows in beam b is given as

$$a^b_k = \{a^b_{1,k}, a^b_{2,k}, \dots, a^b_{F,k}\}, \tag{16}$$

where each element of a^b_k is also an element of $\mathcal{A}$. Herein, an agent has to choose the action from N^F possible actions. This is achieved by constructing the Cartesian product $\mathcal{A} \times F$, where F is the set of flows, $F = \{1, 2, \dots, F\}$. Each element in the Cartesian product represents a pairing of a discrete capacity value $W \in \mathcal{A}$ with a flow $f \in F$. This generates all possible combinations of capacity to be allocated for the available number of flows. To minimize the number of agents and achieve system-wide optimization, one agent can be considered for the entire system. However, this approach comes with a significant drawback: the action space grows exponentially to (N^{FB}), resulting in extremely high training complexity. This makes it impractical for implementation in a flexible CA system within multi-service satellite networks, where dynamic adaptability and computational efficiency are critical.

Both the one agent per flow and one agent per beam models utilize the epsilon-greedy strategy, a commonly employed action selection technique in RL. Under this approach, agents predominantly select the best action with the highest Q-value, optimizing the CA decision based on accumulated knowledge, with a probability of $1-\epsilon$, and introduce an element of randomness by occasionally allocating a random capacity from the action space, with a probability of ϵ as

$$a = \begin{cases} \text{random action}, & \text{for prob. of } \epsilon, \\ \arg\max\limits_{a}\{Q(s,a)\}, & \text{for prob. of } 1-\epsilon, \end{cases} \tag{17}$$

where $Q(s,a)$ is the Q-value corresponding to the action a at state s. This approach applies regardless of whether the model involves one agent per flow or one agent per beam. This dual strategy effectively strikes a balance between exploiting the agent's accumulated experience to maximize immediate rewards and exploring the environment to enhance the learning process over time.

*5) **Reward**:* After taking action, agents receive immediate rewards from the environment, which evaluates how well the allocated capacity meets demand and QoE requirements across all service types. These rewards are calculated based on episodes involving diverse traffic demand scenarios across all service types for the agents to interact with. The goal is to identify a policy that maximizes the expected future rewards based on feedback from these simulated episodes. In the one agent per flow model, during an episode, when the combined capacity needed by the BF flows exceeds the maximum beam capacity, the available capacity is allocated based on service priority. This priority is quantified by a weight vector $\mathbf{p} = (p_1, p_2, ..., p_F)$, indicating the reward penalties imposed on the RL agents. Consequently, during congestion, we assume that the service with the highest tolerance (highest q^f_{QoE}) to wait in a queue is assigned the highest penalty weight. The total reward is then given as a sum of the two sub-rewards. The first sub-reward is expressed in terms of the allocated capacity relative to the available capacity as follows:

$$r^b_{f,k,1} = \begin{cases} 0, & \text{if } \sum_{\forall(b,f)} W^b_f(t) \le W^{\max}, \\ -100 * \frac{p_f}{\sum_{f=1}^{f=F} p_f}, & \text{otherwise}. \end{cases} \tag{18}$$

This reward component penalizes agents if the total allocated capacity exceeds the maximum available capacity $W^{\max}$. The reward of -100 is used to discourage the agents from allocating capacity values that exceed the maximum available capacity for the flows. The penalty per flow is proportional to the priority of the flows. The other sub-reward can be given as a weighted sum of the mean QL and the inverse of allocated capacity depending on requirement satisfaction as

$$r^b_{f,k,2} = \begin{cases} \zeta \bar{q}^b_{f,k} + \frac{\delta}{W^b_{f,k}}, & \text{if } \frac{\varphi^b_{f,k}}{M} \le \bar{P}_{\text{QoE}} \wedge \frac{\Theta^b_{f,k}}{M} \le \bar{P}_{\text{Blk}}, \\ -\bar{q}^b_{f,k}, & \text{otherwise.} \end{cases} \tag{19}$$

This secondary reward component incentivizes the agent to allocate the minimal capacity necessary to achieve and maintain low average QLs, $\bar{q}^b_{f,k}$. Here, ζ and δ are weighting factors that balance the effects of QL and allocated capacity on the reward. The total reward per flow is then given as

$$R^b_{f,k} = r^b_{f,k,1} + r^b_{f,k,2}. \tag{20}$$

Similar to the one agent per flow case, the reward is calculated as a sum of two sub-rewards.

$$r^b_{k,1} = \begin{cases} 0, & \text{if } \sum_{\forall(b,f)} W^b_f(t) \le W^{\max}, \\ -100, & \text{otherwise.} \end{cases} \tag{21}$$

This reward component penalizes agents from making allocation decisions that violate the global capacity constraint. Here, each agent takes a combined action for all flows resulting in different QoE requirement violations and blocking probabilities for each service type. Hence, the second sub-reward is the sum of the rewards from each flow as follows

$$r^b_{k,2} = \sum_{f \in \mathcal{F}} (\zeta \bar{q}^b_{f,k} + \frac{\delta}{W^b_{f,k}}) - \sum_{f \notin \mathcal{F}} \bar{q}^b_{f,k}, \tag{22}$$

where $\mathcal{F} = \left\{ f \mid \frac{\varphi^b_{f,k}}{M} \le \bar{P}_{\text{QoE}} \wedge \frac{\Theta^b_{f,k}}{M} \le \bar{P}_{\text{Blk}} \right\}$. The total reward is then calculated by summing the sub-rewards, given as

$$R^b_k = r^b_{k,1} + r^b_{k,2}. \tag{23}$$

In both the one agent per flow and one agent per beam cases, coordination among agents primarily arises from the shared penalty mechanism and the global capacity constraint, reflecting characteristics of reward shaping and environmental interaction.

B. Deep Q-Network

The DQN utilizes Deep Neural Networks (DNN) to approximate action-value functions for dealing with high-dimensional state space problems, such as flexible CA with time-varying demand, by utilizing the representational power of deep learning. In DQN, each agent creates its own model with two DNNs: the online and the target networks. In each cycle, the agents use the online network to approximate the Q-function $Q(s, a; \omega)$ and choose an action, where ω is the weights of the agent's online network. The target network, with weights $\hat{\omega}$, is used to stabilize the learning process by copying ω after a set number of cycles.

During the training phase, the agent employs the experience replay strategy to enhance convergence speed and solution quality by incorporating a wide range of experiences from different regions of the state space, various actions, and corresponding rewards. By using this method, its transition (s, a, r, s') is stored in the experience replay memory. At each iteration, a random batch of experiences is sampled from this memory to train the learning model. In particular, the application of DQN to solve the problem (8) can be represented as follows: In each learning step (cycle), the agent takes an action of bandwidth allocation after observing its current state. Then it receives a reward from the environment and moves to the next state. After that, its respective experience tuple of (s, a, r, s') is stored in its experience replay memory. A mini-batch of experiences is then sampled to train the online network. Based on that, the parameters of the online network ω are updated to minimize the loss function. The loss function for the one agent per flow configuration is defined as

$$\mathcal{L} = \left(Q' - Q(s^b_{f,k}, a^b_{f,k}; \omega)\right)^2, \tag{24}$$

where Q' is the target Q-value which is computed based on the Bellman optimality principle by adding the reward to the maximum Q-value at the next state as follows:

$$Q' = R^b_{f,k} + \gamma \max_{a'^b_{f,k}} Q(s'^b_{f,k}, a'^b_{f,k}; \hat{\omega}), \tag{25}$$

where γ is the discount factor and $\hat{\omega}$ represent the combination of updated weights and biases in the target network. The Q-value of the online network is updated using the following equation:

$$\begin{aligned} Q(s^b_{f,k}, a^b_{f,k}; \omega) = {} & Q(s^b_{f,k}, a^b_{f,k}; \omega) \\ & + \alpha \left(Q' - Q(s^b_{f,k}, a^b_{f,k}; \omega)\right), \end{aligned} \tag{26}$$

where α is the learning rate, which controls the step size for the update. This equation adjusts the predicted Q-value towards the target Q-value, scaled by the learning rate. Through this iterative process, the parameters of the online network are updated to minimize the loss function, and the online Q-values gradually converge to the optimal Q-values, improving the agent's decision-making ability.

Similarly, for the one agent per beam model, the parameters of the online model ω_b are updated to minimize the loss function as follows

$$\mathcal{L}_b = \left(Q'_b - Q_b(s^b_k, a^b_k; \omega_b)\right)^2, \tag{27}$$

where Q'_b is the target Q-value which is computed as follows:

$$Q'_b = R^b_k + \gamma \max_{a'^b_k} Q_b(s'^b_k, a'^b_k; \hat{\omega}_b), \tag{28}$$

where ω_b is the weight of agent b's online network and $\hat{\omega}_b$ represents the combination of updated weights and biases in the target network of the agent. By minimizing the loss function, the DQN iteratively improves its policy, enabling it to effectively learn optimal actions in complex, high-dimensional environments. After a given number of learning steps, the target network parameters $\hat{\omega}$ and $\hat{\omega}_b$ are updated by copying the values of ω and ω_b. Training continues until convergence. The detailed implementation of the two models is summarized in Algorithm 1.

Remark 1. *Although DQN models have proven their efficiency and effectiveness for resource allocation problems, they sometimes encounter overestimation problems, where the agent consistently selects sub-optimal actions in a given state merely because these actions have the highest Q-value estimates [26].*

Algorithm 1 DQN-BASED CA ALGORITHM

```
1: Initialization:
     • Initialize replay memory $\mathcal{D}$, $\mathcal{D}_b$ .
     • Initialize the online network with random weights $\omega$, $\omega_b$.
     • Initialize the target network with weights $\hat{\omega}$, $\hat{\omega}_b$.
2: for  each episode ( $i = 0$ to max episode) do
3:     Initialize the state $s^b_{f,k}$, $s^b_k$ .
4:     for each cycle ($k = 0$ to $K$) do
5:         Choose an action from the action space using the epsilon-greedy
           method as in (17).
6:         Calculate the total reward according to (20), (23) and observe the
           next state $s'^b_{f,k}$, $s'^b_k$.
7:         Store experiences $(s^b_{f,k}, a^b_{f,k}, R^b_{f,k}, s'^b_{f,k})$ in $\mathcal{D}$
           and $(s^b_k, a^b_k, R^b_k, s'^b_k)$ in $\mathcal{D}_b$.
8:         Take sample minibatch experiences $(s^b_{f,k}, a^b_{f,k}, R^b_{f,k}, s'^b_{f,k})$ from $\mathcal{D}$
           and $(s^b_k, a^b_k, R^b_k, s'^b_k)$ from $\mathcal{D}_b$.
9:         Calculate the target Q_value according to (25) and (28).
10:        Calculate the loss using gradient descent as in (24) and (27).
11:        Update online network parameters $\omega$ and $\omega_b$ to minimize the loss
           function.
12:        Update target network parameters after every $\hat{k}$ ($\hat{k} > 0$) cycles:
           $\hat{\omega} \leftarrow \omega$ and $\hat{\omega}_b \leftarrow \omega_b$.
13:    end for
14: end for
```

Algorithm 2 DDQN/3DQN-BASED CA ALGORITHM

```
1: Initialization:
     • Initialize replay memory $\mathcal{D}'$.
     • Initialize the online network with random weights $\theta$.
     • Initialize the target network with weights $\theta^-$.
2: for each episode ($i = 0$ to max episode) do
3:     Initialize the state ($s^b_k$) for all beams and flows.
4:     for each cycle ($k = 0$ to $K$) do
5:         Choose an action from the action space using the epsilon-greedy
           method as in (17).
6:         Calculate the total reward according to(20), (23) and observe the
           next state $s'^b_k$.
7:         Store experiences $(s^b_k, a^b_k, R^b_k, s'^b_k)$ in $\mathcal{D}'$ .
8:         Sample random minibatch of experiences $(s^b_k, a^b_k, R^b_k, s'^b_k)$ from
           $\mathcal{D}'$.
9:         Select the best action according to (29).
10:        Calculate the target Q-value according to (30).
11:        Perform a gradient descent step to calculate the loss according to
           (31), where the Q-value in the 3DQN model is calculated according
           to (32).
12:        Update online network parameters $\theta$ to minimize the loss function.

13:        Update the target network's parameters after every $\hat{k}$ ($\hat{k} > 0$) cycles:
           $\theta^- \leftarrow \theta$.
14:    end for
15: end for
```

This overestimation occurs because the Q-values predicted by the DQN may not accurately reflect the true expected rewards, leading the agent to make poor decisions. This overestimation problem can be better addressed by using a DDQN.

C. Double Deep Q-Network

DDQN is an improved version of DQN that addresses the issue of Q-value overestimation encountered in DQN by decoupling action selection and evaluation as shown in Fig. 4. Unlike in DQN, where the target network is used for both action selection and evaluation, in DDQN, the online network selects the best action for the next state. In this subsection, we focus exclusively on the one-agent-per-beam DDQN model to reduce the architectural complexity associated with managing a separate agent for each flow. This approach simplifies the overall framework while still leveraging the advanced capabilities of DDQN compared to DQN. Hence, the best action selection is given as

$$a = \arg\max_{a'^b_k} Q\big(s'^b_k, a'^b_k; \theta\big), \tag{29}$$

and the selected action is evaluated by the target network. The target Q-value is then estimated as

$$Q' = R^b_k + \gamma Q\big(s'^b_k, a; \theta^-\big). \tag{30}$$

The gradient descent step can be used to calculate the loss as follows

$$\mathcal{L}' = \Big(Q' - Q(s^b_k, a^b_k; \theta)\Big)^2 \tag{31}$$

The detailed implementation steps of the DDQN-based CA algorithm, which is developed to address problem (8) are summarized in Algorithm 2.

D. Dueling Double Deep Q-Network (3DQN)

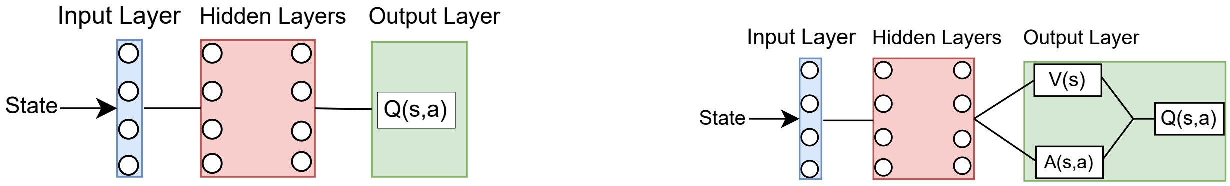
In scenarios where the value of being in a particular state is more significant than the specific actions taken, DDQN may not be efficient due to its inability to estimate state values and action advantages separately [28], [48]. By decoupling the state value and action advantage estimations the 3DQN model helps to reduce Q-value overestimation further and enhance the stability of learning. The Q-value estimation comparison of DDQN and 3DQN is shown in Fig. 5. Here, 3DQN separates the state value $V(s^b_k)$, i.e. the value to be in a particular QL state regardless of the action taken and the action advantage function $A(s^b_k, a^b_k)$, i.e. the advantage of taking a specific action (such as CA) at that specific state for more precise and stable value estimation. We estimate the Q-values by adding the outputs of the state value and advantage values as follows [28], [48]

$$Q(s^b_k, a^b_k; \theta, \theta^S, \theta^A) = V(s^b_k) + A(s^b_k, a^b_k) - \frac{1}{|A_n|} \sum_{a^b_k \in A_n} A(s^b_k, a^b_k), \tag{32}$$

where θ^S, θ^A are the parameters related to the state value function and action advantage function, $|A_n|$ is the number of available actions in the action space and $\frac{1}{|A_n|} \sum_{a^b_k \in A_n} A(s^b_k, a^b_k)$ represents the mean advantage across all possible actions, which is subtracted to normalize the advantage function. The detailed implementation of the CA algorithm is also given in (2), as the only difference with DDQN is in the Q-value estimation given in step 11 of the algorithm.

E. Computational Complexity Analysis

While the overall complexity of the DRL architecture can be influenced by the design of its input, hidden, and output layers, the primary factors are the sizes of the state and action spaces. For the one agent per flow model, where the QL for packets of flow f ranges from 0 to $q_{\max}$, and actions are selected from N discrete options, the complexity scales linearly with the product of the number of beams, flows, and $q_{\max}$. In contrast, for the one agent per beam model, each agent takes actions for multiple flows simultaneously, resulting in a state size of



(a) DDQN.

(b) 3DQN.

Fig. 5: Q-value estimation process in DDQN and 3DQN as in [48].

TABLE III: COMPLEXITY ANALYSIS OF THE PROPOSED METHOD.

Scenario	State Complexity	Action Complexity
1 Agent per Beam	$B \cdot (q_{\text{max}} + 1)^F$	$B \cdot N^F$
1 Agent per Flow	$B \cdot F \cdot (q_{\text{max}} + 1)$	$B \cdot F \cdot N$

$(q_{\text{max}} + 1)^F$ and an action space size of N^F. The complexities of the two scenarios are summarized in Table III.

F. Benchmark Algorithm

This section modifies the algorithm developed in our prior research [47] to obtain a benchmark solution for comparison purposes by assuming Poisson arrivals for all flows. This solution approach utilizes the Lagrangian duality optimization method for CA, as detailed in [47]. It consolidates packet arrivals from all service types into a single queue, which is then served based on a *first-come-first-served* scenario. However, it is worth noting that the CA results obtained by this optimization approach are determined based on perfect knowledge of the arrival rates.

To adapt this benchmark solution to our scheme, we use the predicted traffic demand of an LSTM recurrent neural network as input, as outlined in [49]. To gain a comprehensive understanding of future traffic demand beyond the immediate horizon and to capture longer-term trends more effectively, we employed a multi-horizon LSTM network to predict the input traffic demand of all the TSs of the next cycle for the optimization approach. This approach optimizes forecasting accuracy by concurrently predicting traffic demand for multiple future TSs, enhancing the effectiveness of CA strategies.

IV. PERFORMANCE EVALUATION AND NUMERICAL RESULTS

In this section, we outline the dataset preparation process, discuss the considered hyper-parameter values, present the results obtained using the proposed techniques, conduct a performance comparison, and assess the efficiency of the implemented algorithms.

A. Input Traffic and Traffic Distribution Models

To create diverse services, various distributions of packet arrival rates can be exploited according to the specific services. Common arrival rate distributions encompass the Poisson distribution [50], known for modeling random arrivals, and heavy-tailed distributions [51], [52] which capture the variability and unpredictability often encountered in modern communication networks. We analyze three distinct traffic flows, each corresponding to a different set of characteristics. For the considered 3 service types, Poisson, Pareto (heavy-tailed), and Weibull (heavy-tailed) distributions for estimating the inter-arrival time of packets in the flows. The inter-arrival time for the Poisson distribution denoted τ_{Poi} is calculated from the Cumulative Distribution Function (CDF) of the inter-arrival time given as

$$\textbf{Prob}\left\{\tau_{\text{Poi}} \leq t\right\} = 1 - e^{-\lambda \tau_{\text{Poi}}}, \tag{33}$$

where τ_{Poi} is determined by taking the ratio of the negative natural logarithm of the complement of a random value uniformly sampled between 0 and 1 to the arrival rate of the Poisson-process based flow 1, which is given as

$$\tau_{\text{Poi}} = -\log(1 - R_1)/\lambda_1(t). \tag{34}$$

The CDF of the inter-arrival time of Pareto distribution (τ_{Pa}) is given as [53]

$$\textbf{Prob}\left\{\tau_{\text{Pa}} \leq t\right\} = 1 - (\beta/\tau_{\text{Pa}})^{\eta}, \tag{35}$$

hence, the inter-arrival time of flow 2's packets assumed to follow Pareto distribution is given by

$$\tau_{\text{Pa}} = [\beta/(1 - R_2)]^{(1/\eta)}/\lambda_2(t). \tag{36}$$

Similarly, the CDF of the inter-arrival time of Weibull distribution (τ_{Wei}) is expressed as

$$\textbf{Prob}\left\{\tau_{\text{Wei}} \leq t\right\} = 1 - e^{-\beta_1 \tau_{\text{Wei}}^{\eta_1}}, \tag{37}$$

then, the inter-arrival time of flow 3 packets assumed to follow Weibull distribution is provided by [54],

$$\tau_{\text{Wei}} = \beta_1 \left[-\log(1 - R_3)^{\frac{1}{\eta_1}}\right]/\lambda_3(t), \tag{38}$$

where β and β_1 are the scale parameters, η and η_1 are the shape parameters and R_1, R_2, R_3 are random numbers between 0 and 1. Assuming that the traffic pattern evolves over 24 hours following the trend of the dataset in [55], we generated a random number of packets per second for 10 beams, each supporting 3 distinct service types. To indicate the traffic arrival variability among flows, the Probability Mass Function (PMF) of the packet arrivals for the services is given in Fig. 6.

For the benchmark algorithm, Lagrangian duality, the input traffic was forecasted from historical traffic demand using an LSTM RNN model. To ensure that prediction errors do not affect the comparison, the implemented LSTM model was validated by testing it with a new dataset. After preparing the

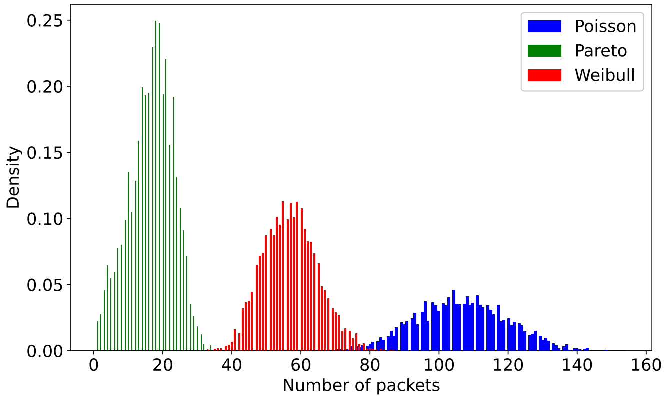


Fig. 6: Probability mass function of the considered flows. Each value on the horizontal axis reflects an arrival rate in a TS, and the corresponding vertical coordinate shows the probability of observing that rate in that TS.

TABLE IV: LSTM HYPER-PARAMETER VALUES.

Parameters	Values
Activation function	tanh
Epochs	100
Forecast horizon	3600
Hidden layers	2
LSTM units	50
Loss function	Mean squared error (MSE)
Number of models	10 (1 per beam)
Sequence length	100

predicted traffic, we employed the $M_t/M_t/1$ queuing analysis for the flow of packets. The hyper-parameters chosen for the LSTM model are summarized in Table IV.

B. Numerical Results and Comparative Analysis

In this section, we analyze and compare the numerical outcomes of various approaches: DQN with one agent per flow (referred to as DQN model 1), DQN with one agent per beam (referred to as DQN model 2), DDQN with one agent per beam, 3DQN with one agent per beam and the benchmark optimization approach utilizing Lagrangian duality. For the DQN, DDQN, and 3DQN models, the agent selects actions from an action space comprising 12 equally spaced values between 0 and the maximum capacity demand of the corresponding flows, generated using the NumPy linspace function. For DQN model 1, the epsilon decay rate (σ) value is 0.9995. If not explicitly specified, the remaining parameter values default to those listed in Table V.

TABLE V: CONSIDERED PARAMETER VALUES.

Parameters	Values
Activation function	Relu, Linear
Cycle duration (M)	1 Hour
Discount factor	0.1
Duration of 1 episode	1 day
Epsilon decay (σ)	0.9998
Loss function	MSE
Experience-replay pool size	50000 [41]
Experience-replay mini-batch size	128
Learning rate	0.01
Maximum QL (q_{max})	2 Mbytes (30 packets) [56]
Maximum capacity (W^{max})	1 Gbps
Normalized packet length	65(KBytes) [57]
Optimizer	Adam [26]
Priority penalty weights	$[0.1, 0.3, 0.6]$
Target update ratio	0.01
Target QoE violation probability ($\bar{P}_{\mathrm{QoE}}$)	0.1 [18], [19]
Target QL (q_{QoE})	$[15, 20, 25]$ packets
The minimal exploration probability	0.001
Target blocking probability ($\bar{P}_{\mathrm{Blk}}$)	0.05 [58]
TS duration (T_{ts})	1 Second

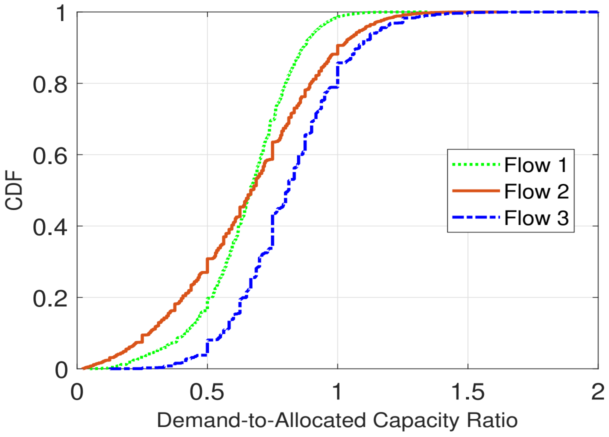
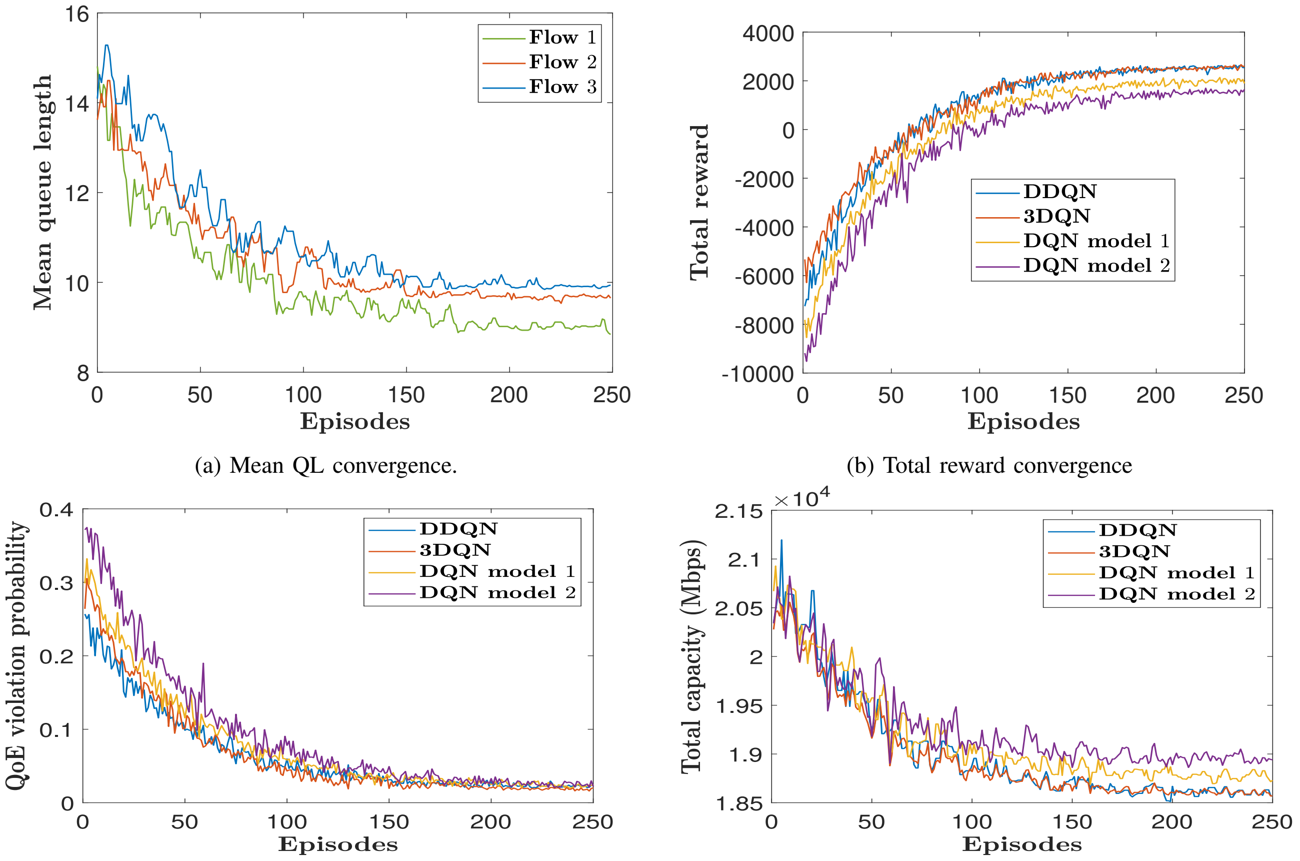


Fig. 7: Demand to capacity ratio of all the flows.

Fig. 7 shows the CDF of the demand-to-allocated capacity ratio. Flow 1 has a higher proportion of samples with lower ratios, indicating less congestion. This reflects its lower priority penalty weight compared to Flows 2 and 3, demonstrating that our CA model effectively prioritizes more critical and delay-sensitive services.

Fig. 8 indicates the convergence of the simulated RL algorithms concerning QL, total reward, probability of QoE violation, and total allocated capacity. Fig. 8a illustrates the convergence of the mean QL over episodes for the three considered flows when using the DDQN method. The figure indicates that the mean QL converges for all flows. Differences in QL values among flows are attributable to the varying target QL requirements and priority weights. Specifically, flow 1 has the lowest target QL and the highest priority weight. This incurs the lowest reward penalty during congestion, leading to the lowest mean QL for this flow. Similarly, Figs. 8b, 8c, and 8d indicate the reward, probability of QoE violation, and total allocated capacity convergence, respectively. The plots reveal that both 3DQN and DDQN achieve similar performance and outperform both the DQN models 1 and 2. This can be attributed to the ability of 3DQN and DDQN to mitigate the overestimation bias commonly encountered in DQN models. The plot also indicates that the DQN model 1 outperforms the DQN model 2 due to the use of individual agents for each flow within every beam. This approach allows each agent to specialize in a specific flow and learn to select the best action from the action space. In contrast, the DQN model 2 assigns one agent per beam, which must manage a larger action space of 12^3 possible actions, making it harder to coordinate multiple flows and resulting in lower performance.

The plots in Fig. 9 indicate the total allocated capacity per cycle against the total traffic demand for all simulated models. To analyze scalability and system performance under varying



(a) Mean QL convergence.

(b) Total reward convergence

(c) Mean QoE violation probability convergence.

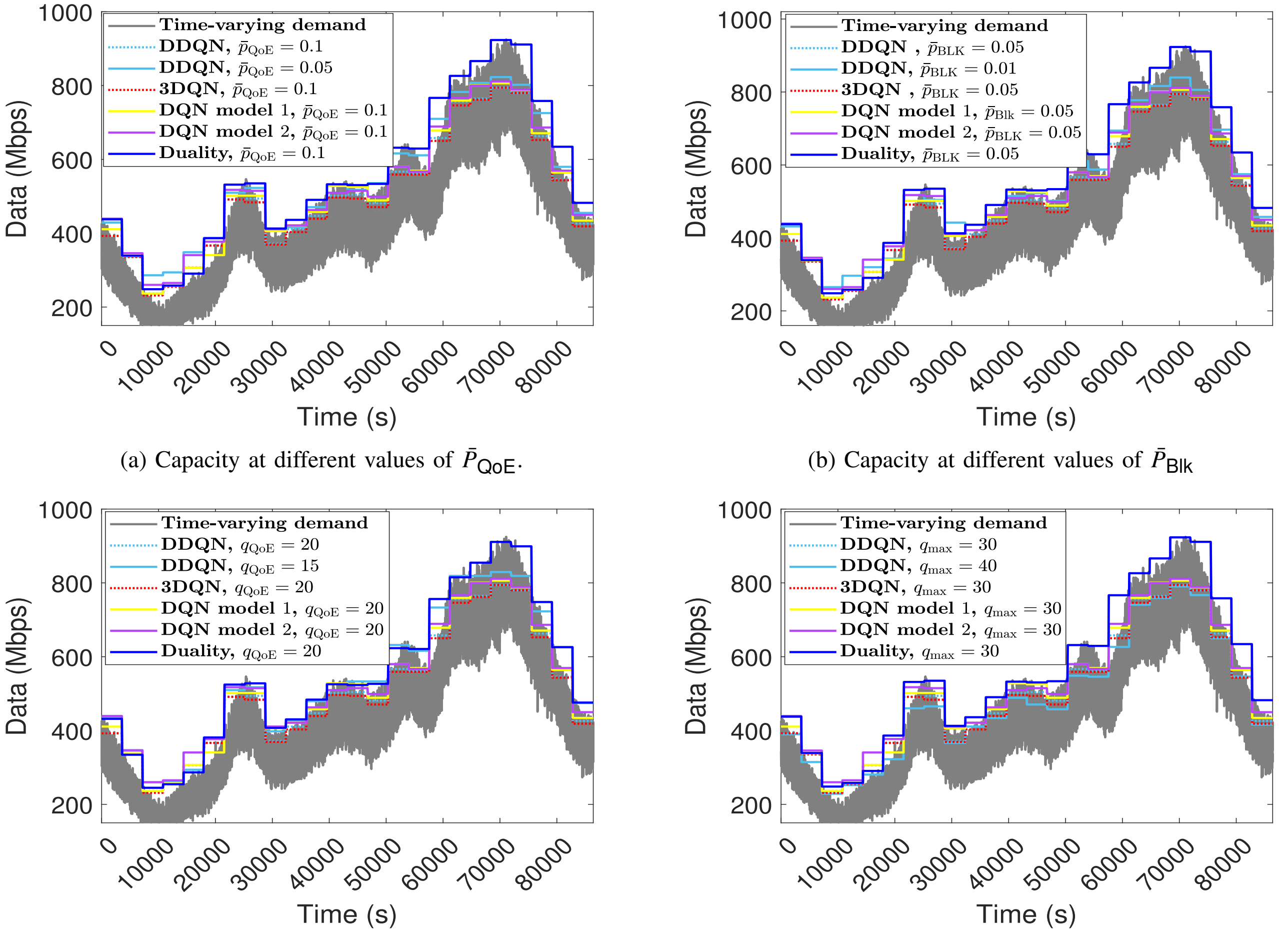
(d) Total allocated capacity convergence

Fig. 8: Convergence plots of the implemented DQN algorithms.

conditions, the total allocated capacity using the proposed DDQN model is evaluated for different values of $\bar{P}_{\mathsf{QoE}}$, $\bar{P}_{\mathsf{Blk}}$, q_{QoE}, and q_{max}, assuming all flows have the same target QL requirement. The total allocated capacity using the DDQN model in Figs. 9a and 9b is given at different values of the target QoE violation ($\bar{P}_{\mathsf{QoE}}$) and target blocking probability ($\bar{P}_{\mathsf{Blk}}$). The plots demonstrate that the system requires allocating a higher capacity for stricter targets, such as a lower QoE violation probability (0.05) compared to a higher value (0.1) and a lower blocking probability (0.01) compared to a higher value (0.05). In addition, the DDQN model results in Figs. 9c and 9d show the total allocated capacity per cycle at different values of target QL (q_{QoE}) and q_{max}. The plots indicate that a CA system intended for services with a higher tolerance for waiting in a queue ($q_{\mathsf{QoE}} = 20$) requires less capacity than a lower tolerance ($q_{\mathsf{QoE}} = 15$) and vice versa. Similarly, a higher value of buffer size ($q_{\mathsf{max}} = 40$) can satisfy the blocking probability requirements with a lower allocated capacity as compared to lower values ($q_{\mathsf{max}} = 30$). This is because a larger buffer size can store more packets that would likely be dropped with a smaller buffer size. However, admitted packets in a system with a large buffer size do not necessarily meet QoE requirements and may lead to buffer bloating.

From the figures we can observe that the 3DQN and the proposed multi-agent DDQN demonstrate a superior performance in allocating less capacity that meets the target QoE requirement compared to the other models. Typically, the Lagrangian duality optimization approach is expected to outperform other methods. However, that approach assumes all arrivals follow a Poisson distribution which fails to accurately capture the traffic flexibility of the diverse traffic patterns of the considered flows. Furthermore, the Lagrangian duality method applied the $M_t/M_t/1$ queueing method which limits the traffic utilization $(\lambda_f^b(t)/\mu_f^b(t))$ value to remain below 1 for all time intervals. This implies that the allocated capacity always exceeds the demand, regardless of the queue state. Such an assumption can lead to an inefficient QL approximation and lower efficiency in CA, as it leaves unutilized capacity to handle any spikes in demand that may exceed the optimal capacity. Consequently, when evaluating the trade-offs between efficiency, adaptability, and scalability, deep learning approaches emerge as promising solutions for QoE-aware dynamic CA in 5G networks.

Figs. 10a and 10b indicate the mean blocking and QoE requirement violation probability per cycle, respectively. The plots demonstrate that all models achieve a value lower than the target value of 0.05 for blocking probability and 0.1 for QoE requirement, indicating their effectiveness. Moreover, these figures, along with Fig. 9, demonstrate that the 3DQN and DDQN methods, which exhibit nearly identical performance, outperform the other methods considered in terms of meeting the target requirements with relatively lower allocated capacity. This is further supported by Fig. 11, which shows, for the same QoE and traffic demand (Mbps), the 3DQN and DDQN models require less capacity, compared to other models, efficiently



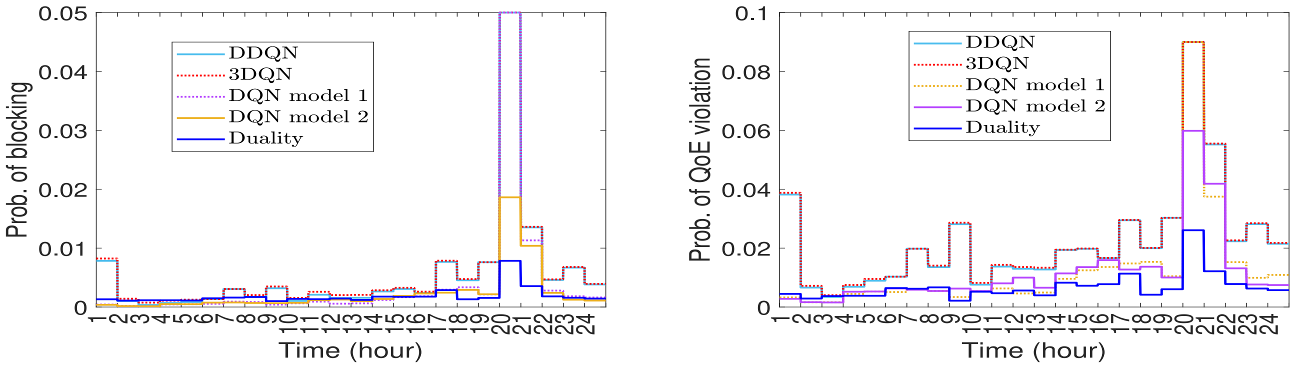
(a) Capacity at different values of $\bar{P}_{\text{QoE}}$.

(b) Capacity at different values of $\bar{P}_{\text{Blk}}$

(c) Capacity at different values of q^f_{QoE}.

(d) Capacity at different values of $q_{\max}$

Fig. 9: Total allocated capacity per cycle and total traffic demand over time.



(a) Mean probability of blocking per cycle.

(b) Mean probability of QoE violation per cycle.

Fig. 10: Probability of blocking and probability of QoE violation.

accommodating a greater number of users.

We can also assess the efficiency of our model by comparing the obtained QoE violation probability results with experimental metrics. These metrics are based on MoS measurements, categorized as a percentage of good or better (%GoB) and poor or worse (%PoW), as described in [18] and [19]. As we do not consider users in our work, we consider the probability of QoE violation per cycle for different samples after the model is trained. As shown in Fig. 12, our model achieves a performance equivalent to the acceptability level of a service with excellent quality, indicated by a MOS above 4.5.

Table VI shows the total runtime of the simulated algorithms on a High Performance Computer (HPC) system using Python 3.8.6 and GCCcore 10.2.0 with 28 CPU cores. From the table, we can observe that the trained DDQN, 3DQN and DQN models have faster inference time compared to the optimization approach, indicating their effectiveness in handling time-varying demands with different arrival rate distributions.

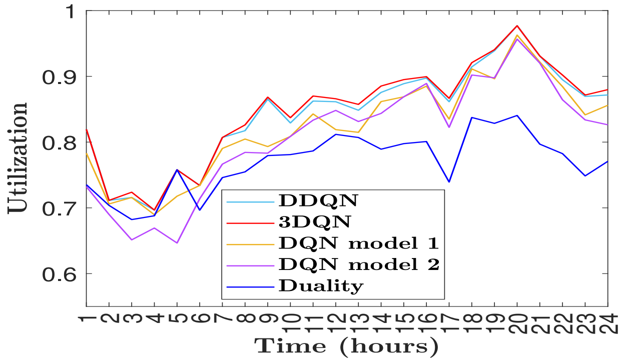


Fig. 11: Capacity utilization efficiency (demand/allocated capacity) of different models.

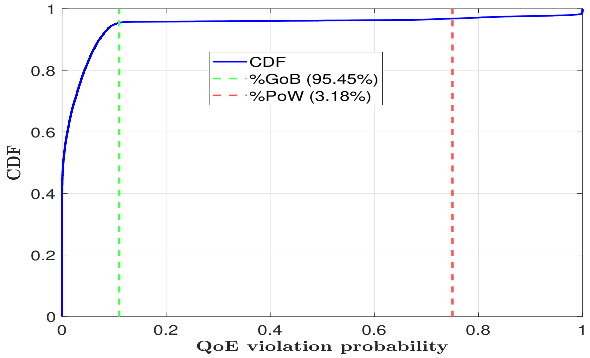


Fig. 12: CDF Plot of QoE Violation Probability: This plot shows the percentage of good or better (%GoB) and poor or worse (%PoW) experiences based on the specified target QoE requirement violation probability of 0.1 for good or better, and the worst 25% (QoE violation probability > 0.75) for poor or worse.

The total time and average time per episode needed for DQN model 1 are higher than the other DRL models. This is because model 1 assumes one agent per flow and requires a total of 30 agents, necessitating a more complex DQN model architecture compared to DQN model 2, 3DQN and DDQN, which assume one agent per beam (10 agents in total). DDQN emerges as a balanced choice, offering a reasonable average time per episode and total convergence time while achieving comparable performance to 3DQN. While 3DQN slightly outperforms DDQN, its higher time complexity makes DDQN a more computationally efficient choice. The increased computational complexity of DDQN compared to DQN model 1 and DQN model 2, is due to the separate action selection and evaluation steps. Similarly, 3DQN introduces higher complexity than DDQN by decoupling the state value and action advantage when calculating Q-values, requiring additional computation to combine them. The scatter plot in Fig. 13 illustrates the relationship between the mean QL and the probability of QoE violation probability, taking the capacity values obtained using DDQN. The results reveal a clear trend: as the mean QL increases, the likelihood of QoE violations also rises. This correlation underscores we can accurately estimate the demand-capacity relationship, allowing for more effective CA strategies.

In summary, DDQN, 3DQN, and DQNs demonstrate comparatively better efficiency in handling various arrival distribution types, optimizing total allocated capacity, and exhibiting inherent adaptability, making them invaluable for meeting the unpredictable demands of dynamic environments as compared to the benchmark method. The main reason for this efficiency is the assumption of different arrival distribution types for all flows in the proposed method, in contrast to the optimization approach that assumes a Poisson process for all flows. Selecting the optimal strategy among the models requires balancing computational resources, system complexity, and the need for tailored decision-making. In our specific simulation, we propose the DDQN model as it shows almost the same performance but lower complexity as compared to 3DQN, and better performance than DQN models 1 and 2.

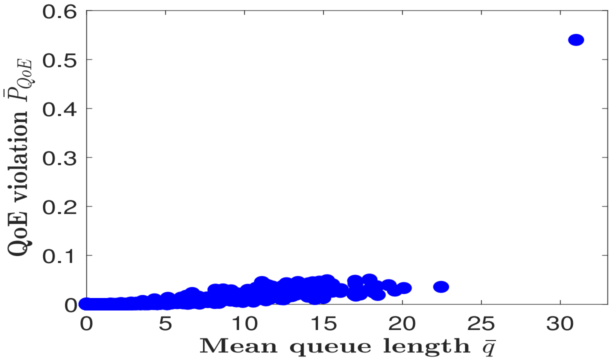


Fig. 13: Mean QL versus Probability of QoE violation.

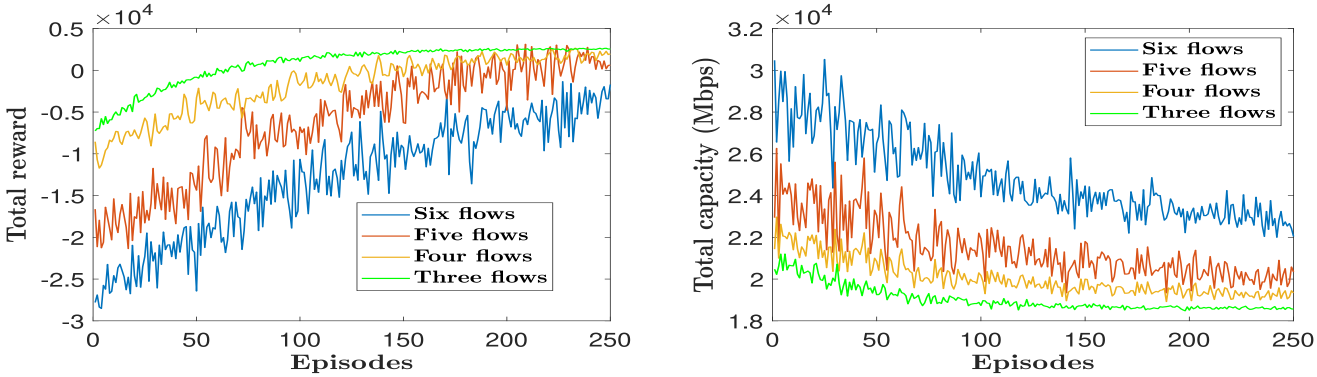
TABLE VI: CONVERGENCE TIME OF THE SIMULATED ALGORITHMS.

Algorithm	Average time per episode (s)	Total convergence time (s)	Episodes to converge
DQN model 1	440.94	110235.755	250
DQN model 2	244.96	61242.086	250
DDQN	273.29	68324.367	250
3DQN	329.85	82463.45	250
Duality	1228.36	122836	100

C. Impact of Increased Number of Flows on Convergence

Managing CA for thousands of service types with QoE-specific requirements poses significant challenges, particularly when considering the computational burden of a large state and action space. To address this, similar service types can be grouped based on QoE requirements, traffic patterns, or priority levels, resulting in fewer traffic flows. The three traffic flow distributions considered in this study, Poisson, Pareto and Weibull represent many existing service types [50]–[52]. However, a substantial number of service groups may follow other distribution types.

In this subsection, we extend our analysis to include additional traffic arrival patterns: Normal, Exponential, and Gamma distributions. These distributions collectively capture a wide range of packet arrival distributions observed in real-world service types. To evaluate the impact of the number of flows, we simulate and compare scenarios with three, four, five, and six flows using a DDQN-based model. Action masking is applied in scenarios with four, five, and six flows to reduce the action space by removing action combinations that exceed the maximum capacity. The convergence results are presented in Fig. 14. The simulations indicate that models with fewer flows (three) converge faster than those with more flows. This



(a) Reward convergence.

(b) Capacity convergence.

Fig. 14: Convergence for different number of flows.

behavior can be attributed to the increased complexity of managing a larger state and action space with additional flows, and the heightened traffic variability resulting from greater variations in arrival distributions. Despite this, the convergence trends are promising, showing an increasing trajectory in the reward and a decreasing trajectory in the capacity convergence. These trends suggest that, although increased flows demand more computational resources and training episodes to reach convergence, the model is capable of achieving optimal performance with further training or the application of complexity-reducing techniques. The results underscore the importance of balancing computational feasibility with the accuracy of CA models in scenarios involving a larger number of traffic flows.

V. Conclusion and Future Work

This paper proposed a QoE-aware flexible CA mechanism, leveraging multi-agent DDQN, that offers significant advancements in optimizing capacity utilization while prioritizing QoE across multiple services. The mechanism exhibits resilience in dynamically adapting to fluctuating traffic demand, ensuring consistent performance and user satisfaction. The simulation results demonstrate that the proposed method enhances the overall capacity utilization efficiency and QoE across various satellite services. Future work may involve accounting for the dynamic nature of LEO constellations and incorporating rain fading alongside mobility, necessitating an agile and responsive strategy for CA. This is essential for efficiently managing communication resources in a rapidly changing SatCom environment. Exploring service prioritization alongside fairness, and load-balancing techniques across multiple LEO SatCom networks to effectively manage time-varying demands and achieve balanced traffic distribution offers a compelling direction for future research.

Acknowledgment

The simulation experiments presented in this paper were carried out by using the University of Luxembourg's HPC facilities https://hpc.uni.lu. We acknowledge the use of AI tools like Grammarly and ChatGPT (https://chat.openai.com/) for assisting with grammar and punctuation corrections.

References

[1] T. M. Kebedew, V. N. Ha, E. Lagunas, D. D. Tran, J. Grotz, and S. Chatzinotas, "Reinforcement learning for qoe-oriented flexible bandwidth allocation in satellite communication networks," in *2023 IEEE Globecom Workshops (GC Wkshps)*, 2023, pp. 305–310.

[2] C.-Q. Dai, M. Zhang, C. Li, J. Zhao, and Q. Chen, "QoE-aware intelligent satellite constellation design in satellite internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4855–4867, 2021.

[3] M. Vincenzi, E. Lopez-Aguilera, and E. Garcia-Villegas, "Maximizing infrastructure providers' revenue through network slicing in 5G," *IEEE Access*, vol. 7, pp. 128 283–128 297, 2019.

[4] M. Irazabal, E. Lopez-Aguilera, I. Demirkol, and N. Nikaein, "Dynamic buffer sizing and pacing as enablers of 5G low-latency services," *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 926–939, 2022.

[5] 3rd Generation Partnership Project (3GPP), "Medium Access Control (MAC) protocol specification," 3GPP, Technical Specification TS 38.321, 2021.

[6] Keysight. (2020) 5G NR protocol structure changes: An overview. [Online]. Available: https://www.keysight.com/blogs/en/inds/2020/07/23/5g-nr-protocol-structure-changes-an-overview

[7] D. Ivanova, E. Markova, D. Moltchanov, R. Pirmagomedov, Y. Koucheryavy, and K. Samouylov, "Performance of priority-based traffic coexistence strategies in 5G mmwave industrial deployments," *IEEE Access*, vol. 10, pp. 9241–9256, 2022.

[8] M. U. Khan, A. Garcia-Armada, and J. J. Escudero-Garzas, "Service-based network dimensioning for 5G networks assisted by real data," *IEEE Access*, vol. 8, pp. 129 193–129 212, 2020.

[9] M. Bosk, M. Gajic, S. Schwarzmann, S. Lange, R. Trivisonno, C. Marquezan, and T. Zinner, "Using 5G QoS mechanisms to achieve qoe-aware resource allocation," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 283–291.

[10] S. Schwarzmann, C. C. Marquezan, R. Trivisonno, S. Nakajima, V. Barriac, and T. Zinner, "ML-based QoE estimation in 5G networks using different regression techniques," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3516–3532, 2022.

[11] R. . S. M. N. Testing, "QoS and QoE in 5G networks Evolving applications and measurements," https://www.itu.int/en/ITU-T/Workshops-and-Seminars/qos/201908/Documents/Jens_Berger_Presentation_1.pdf, 2023, [Online; accessed 13-June-2023].

[12] b. Anil Rama Rao, "Testing 5G Quality of Experience ," https://www.bisinfotech.com/testing-5g-quality-of-experience/, 2023, [Online; accessed 21-June-2023].

[13] R. S. Mogensen and et al, "Empirical IIoT data traffic analysis and comparison to 3GPP 5G models," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021, pp. 1–7.

[14] G. Serazzi, *Impact of Variability of Interarrival and Service Times*. Springer, 2023, pp. 63–72.

[15] B. McCarthy and A. O'Driscoll, "Prediction of inter packet arrival times for enhanced NR-V2X sidelink scheduling," *arXiv preprint arXiv:2311.08227*, 2023.

[16] D. Shi, Y. Xia, and F. Zhang, "Design of buffer queue in low-orbit satellite network based on average route hops," in *International Conference on Intelligent Communication and Networking (ICN)*, 2023, pp. 299–302.

[17] 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.

[18] T. Hossfeld, P. E. Heegaard, M. Varela, and S. Moller, "QoE beyond the MOS: an in-depth look at QoE via better metrics and their relation to MOS," *Quality and User Experience*, vol. 1, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:35445592

[19] N. Zabetian, G. A. Azar, and B. H. Khalaj, "Hybrid non-intrusive QoE assessment of VoIP calls based on an ensemble learning model," *IEEE Transactions on Mobile Computing*, vol. 23, no. 6, pp. 6758–6769, 2024.

[20] A. P. Aguilar, M. Lecci, A. D. Zayas, and H. Wang, "Objective QoE prediction for video streaming services: A novel full-reference methodology," in *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*, 2024, pp. 1–7.

[21] 5G-ACIA, "5G traffic model for industrial use cases," October 2019, white paper. [Online]. Available: https://www.5g-acia.org/wp-content/uploads/2021/04/WP_5G_5G_Traffic_Model_for_Industrial_Use_Cases_22.10.19.pdf

[22] D. Martinez-Mosquera, R. Navarrete, and S. Luján-Mora, "Development and evaluation of a big data framework for performance management in mobile networks," *IEEE Access*, vol. 8, pp. 226 380–226 396, 2020.

[23] P. munoz, n. Adamuz-Hinojosa, J. Navarro-Ortiz, O. Sallent, and J. Perez-Romero, "Radio access network slicing strategies at spectrum planning level in 5G and beyond," *IEEE Access*, vol. 8, pp. 79 604–79 618, 2020.

[24] G. e. a. Fontanesi, "Artificial intelligence for satellite communication and non-terrestrial networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 123–145, 04 2023.

[25] A. Suzuki, R. Kawahara, and S. Harada, "Cooperative multi-agent deep reinforcement learning for dynamic virtual network allocation with traffic fluctuations," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 1982–2000, 2022.

[26] F. G. Ortiz-Gomez, D. Tarchi, R. Martinez, 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.

[27] N. Wang, L. Liu, Z. Qin, B. Liang, and D. Chen, "Capacity analysis of LEO mega-constellation networks," *IEEE Access*, vol. 10, pp. 18 420–18 433, 2022.

[28] D.-D. Tran, S. K. Sharma, V. N. Ha, S. Chatzinotas, and I. Woungang, "Multi-agent DRL approach for energy-efficient resource allocation in URLLC-enabled grant-free NOMA systems," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 1470–1486, 2023.

[29] D. Lim and I. Joe, "MAARS: Multiagent actor–critic approach for resource allocation and network slicing in multiaccess edge computing," *Sensors*, vol. 24, no. 23, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/23/7760

[30] N. D. L. Fuente and D. A. V. Guerra, "Comparative study of deep reinforcement learning models," 2024. [Online]. Available: https://arxiv.org/abs/2407.14151

[31] H. Jia, Y. Wang, H. Peng, and W. Li, "Dynamic beam hopping and resource allocation for non-uniform traffic demand in NGSO satellite communication systems," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 1, pp. 816–830, 2025.

[32] S. Zhang, R. Chai, C. Liang, and Q. Chen, "Dynamic resource allocation for multibeam satellite communication systems," *IEEE Internet of Things Journal*, vol. 11, no. 22, pp. 36 907–36 921, 2024.

[33] R. Chai, G. Yang, L. Liu, and Q. Chen, "DRL-based dynamic resource allocation for multi-beam satellite systems," *IEEE Transactions on Network and Service Management*, vol. 21, no. 4, pp. 3829–3845, 2024.

[34] T. Van Chien and et al., "User scheduling and power allocation for precoded multi-beam high throughput satellite systems with individual quality of service constraints," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 1, pp. 907–923, 2023.

[35] S. Yuan, Y. Sun, M. Peng, and R. Yuan, "Joint beam direction control and radio resource allocation in dynamic multi-beam LEO satellite networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 6, pp. 8222–8237, 2024.

[36] K. S. Wheatman, F. Mehmeti, M. Mahon, and T. L. Porta, "QoE-analysis of 5G network resource allocation schemes for competitive multi-user video streaming applications," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–6.

[37] N. Eswara, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "Perceptual QoE-optimal resource allocation for adaptive video streaming," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 346–358, 2020.

[38] Z. Wang, X. Liu, H. Gu, S. Mao, and Z. Peng, "QoE-aware bandwidth resource allocation strategy for ultra-high-definition video services in B5G: A game theoretic approach," *IEEE Internet of Things Journal*, vol. 12, no. 6, pp. 7564–7576, 2025.

[39] F. Rahdari, M. R. Khayyambashi, and N. Movahhedinia, "A QoE-aware nonlinear fuzzy radio resource management approach for revenue enhancement," *IEEE Systems Journal*, vol. 17, no. 1, pp. 1407–1418, 2023.

[40] N. Zabetian and B. H. Khalaj, "QoE-aware network pricing, power allocation, and admission control," *IEEE Transactions on Mobile Computing*, vol. 23, no. 3, pp. 2231–2240, 2024.

[41] Z. Li, Z. Xie, and X. Liang, "Dynamic channel reservation strategy based on DQN algorithm for multi-service LEO satellite communication system," *IEEE Wireless Communications Letters*, vol. 10, no. 4, pp. 770–774, 2021.

[42] P.-Y. Su, K.-H. Lin, Y.-Y. Li, and H.-Y. Wei, "Priority-aware resource allocation for 5G mmwave multicast broadcast services," *IEEE Transactions on Broadcasting*, vol. 69, no. 1, pp. 246–263, 2023.

[43] I. AlQerm and J. Pan, "Deepedge: A new QoE-based resource allocation framework using deep reinforcement learning for future heterogeneous edge-IoT applications," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 123–134, 2023.

[44] Y. Fu, X. Wang, and F. Fang, "Multi-objective multi-dimensional resource allocation for categorized QoS provisioning in beyond 5G and 6G radio access networks," *IEEE Transactions on Communications*, vol. 72, no. 3, pp. 1790–1803, 2024.

[45] P. Oliver-Balsalobre, M. Toril, S. Luna-Ramírez, and R. G. Garaluz, "Self-tuning of service priority parameters for optimizing quality of experience in LTE," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3534–3544, 2018.

[46] P. Oliver-Balsalobre, M. Toril, S. Luna-Ramirez, and J. M. R. Avilés, "Self-tuning of scheduling parameters for balancing the quality of experience among services in LTE," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 7, 2016. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-015-0508-x

[47] T. M. Kebedew, V. N. Ha, E. Lagunas, J. Grotz, and S. Chatzinotas, "QoE-aware cost-minimizing capacity renting for satellite-as-a-service enabled multiple-beam satcom systems," *IEEE Transactions on Communications*, pp. 1–1, 2023.

[48] M. Sewak, "Deep Q Network (DQN), Double DQN, and Dueling DQN," *Deep Reinforcement Learning*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:198329027

[49] M. A. Tariq, M. M. Saad, M. Ajmal, A. Siddiqa, J. Seo, Y. Haishan, and D. Kim, "Network slice traffic demand prediction for slice mobility management," in *2024 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2024, pp. 281–285.

[50] P. Korrai, E. Lagunas, S. K. Sharma, S. Chatzinotas, A. Bandi, and B. Ottersten, "A RAN resource slicing mechanism for multiplexing of eMBB and URLLC services in OFDMA based 5G wireless networks," *IEEE Access*, vol. 8, pp. 45 674–45 688, 2020.

[51] T. M. Tatarnikova and O. Kutuzov, "Determination of the buffer capacity the network node when servicing self-similar traffic modeled by the weibull distribution," in *10th Mediterranean Conference on Embedded Computing (MECO)*, 2021, pp. 1–4.

[52] S. R. Pandey, M. Alsenwi, Y. K. Tun, and C. S. Hong, "A downlink resource scheduling strategy for URLLC traffic," in *IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2019, pp. 1–6.

[53] Z. Harpantidou and M. Paterakis, "Random multiple access of broadcast channels with pareto distributed packet interarrival times," *IEEE Personal Communications*, vol. 5, no. 2, pp. 48–55, 1998.

[54] P. Bhattacharya and R. Bhattacharjee, "A study on weibull distribution for estimating the parameters," *Journal of Applied Quantitative Methods*, vol. 5, no. 2, pp. 234–241, 2010.

[55] 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.

[56] U. Speidel and L. Qian, "Striking a balance between bufferbloat and TCP queue oscillation in satellite input buffers," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.

[57] X. Xu, Q. Wang, C. Liu, and C. Fan, "A satellite network data transmission algorithm based on adaptive lt code," in *2021 International Conference on Space-Air-Ground Computing (SAGC)*, 2021, pp. 100–105.

[58] 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.