SCAN: ML-based Slice Congestion and Admission Network Controller

Abida Perveen, Berna Bulut Cebecioglu, Raouf Abozariba, Mohammad Patwary,

Adel Aneiba, Anish Jindal, M. Omar Al-Kadri

***Abstract*—Network slicing enables 5G/6G networks to support Ultra-Reliable Low-Latency Communication (ULLC), enhanced Mobile Broadband (eMBB) and Massive Machine-Type Communication (mMTC). However, while this virtual networking technology enhances network efficiency, it also adds substantial signaling overhead. Maintaining sub-millisecond latency and managing dense deployments require continuous signaling at high resolution, which keeps hardware components active, leading to increased energy consumption. In this paper, we introduce a novel network controller that manages slice congestion and admission, designed to meet flexible Quality-of-Experience requirements for both priority and non-priority traffic. Utilizing metadata from Internet of Things (IoT) device applications and network characteristics, we introduce adaptability and elasticity features, enabled by transfer and reinforcement learning, significantly lowering signaling overhead and network resources. Further, analytical results show the proposed framework effectively reduces rejection rates and congestions across varying mMTC and eMBB traffic loads.**

***Index Terms*—5G/6G, transfer learning, reinforcement learning, admission control, and energy consumption.**

I. Introduction

The increasing demand for supporting heterogeneous traffic in 5G networks added complexity and overhead, leading to congestion and poor resource allocation (RA) management [1]. To manage increasing network complexity, 3GPP mandated network slicing, which partitions networks into separate functional segments. This approach enables a single physical network to be logically partitioned into multiple virtual networks, using either exclusive or shared network resources across the edge-to-edge chain (also known as network instances) [2]–[4]. These virtual networks are tailored to accommodate the diverse requirements of emerging applications [5]. Based on QoE parameters, 3GPP classifies traffic into three key network slices: ultra-reliable low-latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC) [6], [7]. These service-specific slices are further categorized into hard, guaranteed-soft (GS), and best-effort (BE) QoE traffic [8]. Effective admission control (AC) and optimal resource management–including allocation and utilization–depend heavily on the ability to strategically select and implement network slice instances, guided by service QoE demands and real-time network load conditions [9], [10]. In the event a network becomes overloaded, incoming requests are placed in a queue pending slice admission, where they are prioritized based on GS and BE QoE demand classification. To reduce delays, the slice queue's capacity is typically capped. However, if the number of slice requests surpasses this limit, congestion occurs, and requests that can not be queued are immediately dropped, leading to poor user experience and high energy consumption [11], [12].

3GPP introduced a network slice management and orchestration function within the 5G architecture [2], [13]. However, this framework primarily outlines design principles and interface guidelines, leaving implementation details to vendors. Building on this reference architecture, [14] introduced a Mobile Virtual Network Operator. This approach reallocates resources from low to high-priority slices, aiming to reduce rejection rates under overloaded network conditions. However, such reactive reconfigurations also incurs multiple signaling exchanges, adding core network overhead and congestion [15], [16].

Machine learning (ML) solutions were frequently explored in the literature to enhance network slice management and orchestration [17], [18]. Conventional ML methods ineffectively address wireless network challenges due to cloud-based, data-intensive processing that increases network latency and congestion. Advanced techniques such as FL and DRL generate extensive data exchanges overhead and are computationally demanding, limiting their effectiveness in latency-sensitive network environments and incur greater energy consumptions [19]–[21].

More recently, transfer learning (TL) has emerged as a promising solution, accelerating learning while minimizing redundant data transmission [22]. In addition, Reinforcement Learning (RL) have been applied to wireless network management [23]–[27], but current approaches suffer from scalability and computational complexity limitations. Recent research highlights the need for more efficient solutions, with Transfer Learning (TL) emerging as a promising approach to reduce computational demands and data offloading [22], [28]–[30], though no existing research has fully explored TL's potential to mitigate network bottlenecks. To this end, we propose a novel framework that minimizes control signaling through efficient slice management, combining a simple yet effective learnable clustering module and transfer learning techniques.

A. Perveen, B. B. Cebecioglu, R. Abozariba, and A. Aneiba are with the School of Computing and Digital Technology, Birmingham City University, United Kingdom (emails: {abida.perveen, berna.bulut, raouf.abozariba, adel.aneiba}@bcu.ac.uk).

M. Patwary is with the School of Mathematics and Computer Science, University of Wolverhampton, United Kingdom (email: patwary@wlv.ac.uk).

A. Jindal is with the Department of Computer Science, Durham University, United Kingdom (email: anish.jindal@durham.ac.uk).

M. O. Al-Kadri is with the college of computing and information technology, University of Doha for Science and Technology, Doha, Qatar (email: omar.alkadri@udst.edu.qa).

In summary, we make the following key contributions:

- We introduce a novel unified framework for network slice admission control, combining unsupervised clustering with reinforcement learning. Our approach dynamically adapts to network conditions by first clustering similar slice requests using learned representations, followed by employing a policy gradient algorithm that optimizes admission decisions in the presence of heterogeneous traffic classes. We successfully prove that this hierarchical approach achieves near-optimal slice acceptance rates while maintaining global fairness and reducing energy usage through efficient resource allocation.
- We develop an adaptive resource allocation mechanism that jointly optimizes both intra-slice and inter-slice resource distribution. Our key contribution is a novel unified cost estimation function that mathematically guarantees fair resource distribution while dynamically adjusting slice elasticity based on real-time demand. Experimental results show our approach substantially reduces service rejection rates under high-load conditions compared to state-of-the-art baselines, while consistently maintaining QoS guarantees across heterogeneous traffic patterns.
- We demonstrate that intelligent clustering techniques, integrable with existing 3GPP standards, significantly reduce signaling overhead in network slice operations. This approach decreases energy consumption while maintaining excellent baseline service quality. Our solution requires no hardware modifications and can be deployed as a software upgrade to existing infrastructure, offering immediate energy efficiency benefits for mobile network operators.

This article is organized as follows. The proposed slice congestion and admission network controller framework, its design, and statistics are explained in detail in Section II. Section III 'Performance Analysis and Results' offers a comparative characterization based on previously conducted research. Finally, the concluding remarks on the proposed research are presented in Section IV.

II. System Model

Effectively managing heterogeneous traffic while ensuring proportional slice capacity and low latency remains a challenge in future networks [10], [31]. We propose a network controller (SCAN) that manages congestion and admission, aligned with current industry standards [2], [5]. It has three main parts: a system to analyze demand, a system to cluster and queue requests, and a controller to manage resources and admission, as shown in Fig. 1. This design is applicable to both traditional and cloud-based networks.

The network slice selection process begins during user equipment (UE) registration. Upon powering on, the UE transmits an admission request and Registration message (containing user ID and service type) to the connected gNodeB in the Radio Access Network (RAN). A demand analyzer classifies the requested slice's QoE requirements—distinguishing GS from BE QoE using a classification mask. ML and optimization-driven clustering in the DCQ system then group similar requests by service type and QoE, queuing them for admission.

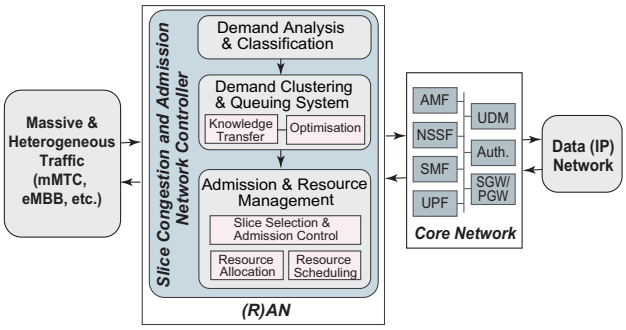

Fig. 1. Architecture of the Slice Congestion and Admission Network Controller (SCAN).

The gNodeB routes clustered requests to the Access and Mobility Management Function (AMF). The AMF validates the UE's service authorization by retrieving subscription data from the Unified Data Management (UDM). After authentication, the Network Slice Selection Function (NSSF) assigns a slice ID, which includes shared Network Function (NF) instances for users in the same cluster. The default slice stores this metadata in the Unstructured Data Storage Function (UDSF) before forwarding the request to the Session Management Function (SMF) and Serving/Packet Gateway (SGW/PGW) for connection setup and data routing [6], [13].

Network setup: Our network consists of multiple slices, defined as $\mathcal{S} = \{1, 2, \ldots, S\}$. The network includes M mMTC and N eMBB devices. The mMTC users are denoted as $\mathbf{U}_{MTC} = \{\mathbf{u}_{best}^{p_1}, \mathbf{u}_{soft}^{p_1}\}$, while eMBB users are defined as $\mathbf{U}_{MBB} = \{\mathbf{u}_{best}^{p_2}, \mathbf{u}_{soft}^{p_2}\}$, where p_1 and p_2 indicate their respective categories. It is assumed that, $\mathbf{u}_{best}^{p_1} = \{1, 2, \ldots, \kappa\}$ and $\mathbf{u}_{soft}^{p_1} = \{\kappa + 1, \kappa + 2, \ldots, M\}$, and $\mathbf{u}_{best}^{p_1} \cap \mathbf{u}_{soft}^{p_1} = \emptyset$. Likewise, $\mathbf{u}_{best}^{p_2} = \{1, 2, \ldots, \iota\}$ and $\mathbf{u}_{soft}^{p_2} = \{\iota + 1, \iota + 2, \ldots, N\}$, and $\mathbf{u}_{best}^{p_2} \cap \mathbf{u}_{soft}^{p_2} = \emptyset$. The demands exhibit varying characteristics, defined as $\mathcal{J} = \{1, 2, 3, \ldots, J\}$, with values ranging between j_{min} and j_{max}, where $j \in \mathcal{J}$. These predefined characteristics are stored in the AMF repository [32] and are used to classify slice admission requests accordingly. Each device may connect to K heterogeneous slices from $\mathcal{S}$, denoted as $\Lambda = \{1, 2, 3, \ldots, K\}$, though for simplicity, we assume $K = 1$. Key symbols are listed and described briefly in Table I. Fig. 2 illustrates the systematic diagram of the proposed model, which is further detailed in the following subsections.

A. Demand Analysis and Classification

A user $u \in \mathbf{U}_{MTC}$ seeking access to the s^{th} slice with specific QoE requirements submits a request in the form of $\mathbf{a}_u$, which is stored in the corresponding demand matrix $\mathbf{A}$ (as shown in (1)) in the 5G slice controller's repository within the RAN. The vector $\mathbf{a}_u$ encapsulates user-centric application parameters and can be written as:

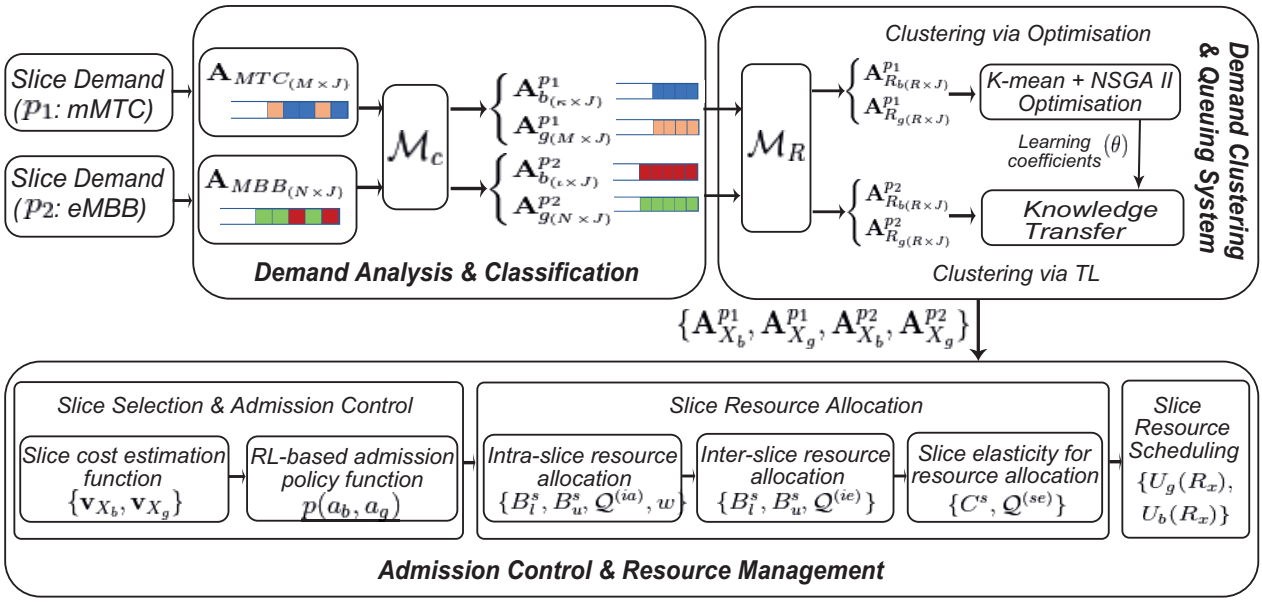

Fig. 2. SCAN system diagram for capacity and delay optimization.

TABLE I
KEY SYMBOLS AND DEFINITIONS

Symbols	Definitions
$\mathbf{U}_{MTC}$,$\mathbf{U}_{MBB}$	Set of users belonging to mMTC and eMBB
$\mathbf{u}^{p_1}_{best}$, $\mathbf{u}^{p_2}_{best}$	Set of users belonging to best-effort demand of mMTC and eMBB, respectively.
$\mathbf{u}^{p_2}_{soft}$, $\mathbf{u}^{p_2}_{soft}$	Set of users belonging to guaranteed soft demand of mMTC and eMBB, respectively.
$\mathcal{S}$	Set of slices in the network
$\mathbf{A}_{MTC}$,$\mathbf{A}_{MBB}$	Demand matrix of mMTC and eMBB
$\mathcal{M}_c$, $\mathcal{M}_R$	Demand classification masks and ranking masks
C^s_{que}, C^s_{req}	Slice queuing capacity and required capacity
$D_{(x)}$, $\mathcal{D}_{(x)}$	Queue waiting time and threshold time of cluster x
$d_{(x_u)}$	uth request waiting time from cluster x
$\mathbf{v}$, $\mathbf{w}$	Cost estimation function and Network weights for slice selection
$p_{(a_b,a_g)}$	RL-based admission control policy function
B^s_l, B^s_u	Slice lower and upper configuration bounds
$\mathcal{Q}$	Number of rejected requests
α_x	Admission of cluster x to slice s
w_b, w_g	Acquired reward on BE and GS demand admission
$C^s_{res\_sig}$	Slice reserved uplink signaling capacity
$C^s_{exp\_sig}$	Experienced signaling capacity
$C^s_{b\_sig}$, $C^s_{g\_sig}$	Slice signaling capacity for BE and GS demand
$C^s_{cls\_sig}$	Signaling capacity after clustering
$C^s_{b\_cls\_sig}$	Clustered BE signaling capacity
$C^s_{g\_cls\_sig}$	Clustered GS signaling capacity
L	Number of requests within a cluster
t_u	Aggregate waiting time for admission requests in cluster
$U(R_x)$	xth cluster utility
U^s, U	sth slice utility and Network utility on set $\mathcal{S}$

$$\mathbf{A}_{MTC} = \left[a_{(m,j)}\right]_{M\times J}. \tag{1}$$

For eMBB requests, we create a demand matrix $\mathbf{A}_{MBB}$ of size $[N \times J]$. The SDAC system processes each request $\mathbf{a}_u$ (whether mMTC or eMBB) using a classification mask function $\mathcal{M}_c$ that separates traffic into BE and GS-QoE categories. This classification helps maximize the acceptance ratio of slice requests through clustering, governed by the membership function:

$$\mathcal{M}_c = \mathbb{1}(\mathbf{a}_u \in \mathbf{u}_{best})c_b + \mathbb{1}(\mathbf{a}_u \in \mathbf{u}_{soft})c_g\,. \tag{2}$$

where, c_b and c_g classifiers for BE and GS-QoE demand, respectively. $\mathbb{1}(\cdot)$ is the indicator function. When a slice request for a specific service type is received, the slice controller evaluates each characteristic value of the request. Based on the analysis, the controller classifies the request and assigns it to the user-centric row in either the BE ($\mathbf{A}^{p_1}_b$) or the GS-QoE demand matrix ($\mathbf{A}^{p_1}_g$) for the respective p_1, as shown below:

$$\mathbf{A}^{p_1}_b = \left[b_{(i,j)}\right]_{\kappa\times J}, \quad \mathbf{A}^{p_1}_g = \left[g_{(i,j)}\right]_{(i>\kappa)\times J}. \tag{3}$$

Similarly, for users belonging to eMBB, the BE and GS-QoE-based demand matrices are constructed, denoted as $\mathbf{A}^{p_2}_{b(\iota\times J)}$ and $\mathbf{A}^{p_2}_{g(N\times J)}$, respectively. The QoE-based slice service demands are then forwarded to the DCQ system for processing. This step aims to minimize redundancy in the admission request signaling, which could otherwise lead to congestion and resource starvation in the network.

B. Demand Clustering and Queuing System

Fig. 3 illustrates the key SCAN operations for slice request clustering. The DCQ system employs K-means clustering

and ranking algorithms to optimize slice acceptance rates by identifying and reducing redundant requests. Details of this process are provided in the following subsections.

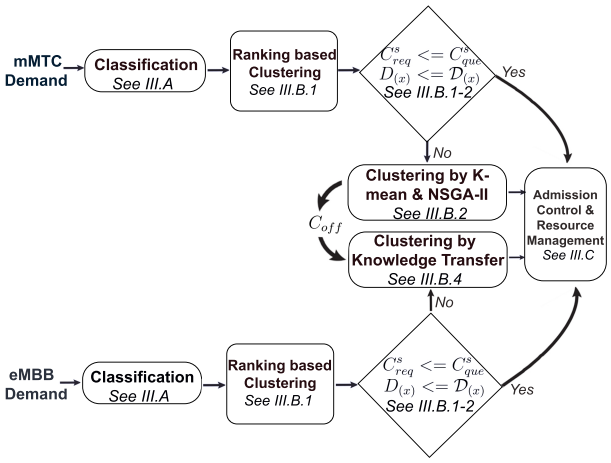

Fig. 3. Simplified strategic flow diagram for clustering by optimization and knowledge transfer.

1) Resource Clustering Optimization: The slice queuing capacity, denoted as C^s_{que}, is constrained to prevent requests that generate excessive delays. Under normal conditions, the required capacity for incoming admission requests C^s_{req}, scales with the queuing capacity, resulting in lower waiting times. However, as service demand increases, the required capacity grows exponentially [11], [18]. This can lead to bottleneck congestion at the network edge, causing higher slice request rejection rates and reducing operator revenue and network QoS due to inefficient core network resource use. We formulate this as an optimization problem, aiming to manage admission requests—whether BE or GS-QoE requests from MTC or p_1 users, to reduce edge-level rejections (α) through efficient queue management. We express this as:

$$\begin{aligned} &\min \sum_{i=b_{(1)}}^{b_{(\kappa)}} \alpha_{(i)} + \sum_{i=g_{(\kappa+1)}}^{g_{(M)}} \alpha_{(i)} \\ &\text{s.t.} \quad \sum_{i=b_{(1)}}^{b_{(\kappa)}} C^s_{\text{req}}(i) + \sum_{i=g_{(\kappa+1)}}^{g_{(M)}} C^s_{\text{req}}(i) \le C^s_{\text{que}}, \\ &\qquad \sum_{u=1}^{M} \beta_{(u)} \le 1. \end{aligned} \tag{4}$$

The total capacity requested by slice admission requests, C^s_{req}, should not surpass the total reserved slice queuing capacity across M mMTC slice requests. An admission index β of 1 indicates that the request from the uth user is accepted into the queue, otherwise zero. All requests from the set $\mathbf{U}_{MTC}$ must be accepted for queuing by the RAN controller.

This work applies ranking-based clustering to reduce slice request rejection. This method efficiently computes similarity within clusters [33]. Upon receiving a request, the DCQ system compares it with existing ones based on homogeneous demand characteristics. Requests from BE or soft-QoE demand matrix of p_1, pass through the ranking-based clustering mask $\mathcal{M}_R$, transforming QoE demand matrices ($\mathbf{A}^{p_1}_b$ and $\mathbf{A}^{p_1}_g$) into Ranked-based versions ($\mathbf{A}^{p_1}_{R_b}$ and $\mathbf{A}^{p_1}_{R_g}$), ensuring $b_R \le b_\kappa$ and $g_R \le g_M$. Similarly, requests from p_2 users are passed through the mask $\mathcal{M}_R$, and $\mathbf{A}^{p_2}_{R_b}$ and $\mathbf{A}^{p_2}_{R_g}$ are constructed, where $b_R \le b_\iota$ and $g_R \le g_N$. Now, C^s_{req} acquired by the clustered requests would not exceed the overall reserved slice queuing capacity over a set of users from either mMTC or eMBB as follows:

$$\sum_{i=1}^{b_R} \phi_{p_1} C^s_{req}(i) + \sum_{i=\kappa+1}^{g_R} \phi_{p_1} C^s_{req}(i) \le C^s_{que}, \tag{5}$$

$$\sum_{i=1}^{b_R} \phi_{p_2} C^s_{req}(i) + \sum_{i=\iota+1}^{g_R} \phi_{p_2} C^s_{req}(i) \le C^s_{que}, \tag{6}$$

where ϕ_{p_1} and ϕ_{p_2} represent the admission indicators for p_1 (mMTC) and p_2 (eMBB) users, respectively, within the ranking-based clustering process. ϕ_{p_1} and ϕ_{p_2} are 1 for p_1 and p_2 user requests, respectively; otherwise, 0. These flags activate the relevant ranked demand matrices in capacity calculation.

2) Latency-Aware Demand Clustering: As noted earlier, requests must be clustered such that their delay (or waiting time) does not exceed a threshold, ensuring the agreed QoS is maintained [34]. Minimizing delay within clusters can be formulated as an optimization problem, with the objective of clustering requests in a way to minimize delay $D_{(x)}$, in the slice queue, and reduce rejection rate. This is mathematically described as follows:

$$\begin{aligned} &\min D_{(x)}, \\ &\text{s.t.} \quad \sum_{x_u=1}^{L} d_{(x_u)} \le \mathcal{D}_{(x)}, \\ &\qquad \sum_{u=1}^{M}\sum_{x=1}^{X} \beta_{(u,x)} \le |\mathbf{U}_{\text{MTC}}|. \end{aligned} \tag{7}$$

The aggregate waiting time, t_u, for admission requests in cluster x must satisfy $\sum_{u\in\mathbf{U}_x} t_u \le \mathcal{D}_x$, where $\mathbf{U}_x$ denotes requests in cluster x. The admission indicator is defined as:

$$\beta_{u,x} = \begin{cases} 1 & \text{if } u \in \mathbf{U}_{\text{MTC}} \text{ admitted to cluster } x, \\ 0 & \text{otherwise,} \end{cases}$$

where clusters partition $\mathbf{U}_{\text{MTC}}$ by homogeneous QoS requirements.

Minimizing request rejection rate at the network edge, caused by long delays and limited capacity, is an NP-hard problem. To address this, optimization and ML-based techniques are explored to solve (4) and (7) simultaneously. User requests are grouped into R clusters based on homogeneous slice-service demand using a ranking-based approach.

K-means clustering aims to minimize the sum of the squared distances between data points to their closest centers. It offers low computational overhead and facilitates fast convergence, meeting real-time networking demands. By partitioning user requests into homogeneous groups, K-means effectively reduces the dimensionality and variability of the optimization problem. This reduction enables more targeted and efficient

resource allocation, thereby mitigating state-space complexity and accelerating learning processes in reinforcement-based algorithms. In contrast, NSGA-II (Non-dominated Sorting Genetic Algorithm II) is particularly effective in addressing multi-objective optimization problems involving conflicting goals, such as minimizing latency in mMTC while maximizing bandwidth efficiency for eMBB services. NSGA-II guarantees rapid convergence to a diverse set of Pareto-optimal solutions, making it particularly suitable for cross-slice resource allocation in dynamic service environments. Together, these techniques enable scalable and adaptive optimization within the SCAN framework, achieving efficient performance with minimal computational overhead.

Given the high volume of connectivity requests, optimization is applied to p_1 requests, and the knowledge gained is transferred to p_2 requests as coefficients to accelerate the admission control, as illustrated in Fig. 3. The key step in this approach is defining a suitable genetic representation of the requests from set either $\mathbf{u}_{best}^{p_1}$ or $\mathbf{u}_{soft}^{p_1}$. The goal is to optimally group L requests within cluster x so that the total delay (or waiting time) $D_{(x)}$ of L requests does not exceed the threshold $\mathcal{D}_{(x)}$. This is achieved by efficiently scheduling each request u of p_1 in the cluster x based on the minimum aggregate waiting time:

$$D_{(x)} = \sum_{x_u=1}^{L} d_{(x_u)} \leq \mathcal{D}_{(x)} \,, \tag{8}$$

where, $d_{(x_u)}$ represents the delay experienced by request u in cluster x. Using the M/M/1 queuing model [35], the delay $d_{(x_u)}$ in the proposed model can be expressed as follows:

$$d_{(x_u)} = \frac{1}{(\mu - L)} - \frac{1}{\mu} \,, \tag{9}$$

where, μ denotes the mean rate of the request execution from clusters, while L represents the request arrival rate within cluster x. After optimization, the demand clustering metric, either $\mathbf{A}_{X_b}^{p_1}$ or $\mathbf{A}_{X_g}^{p_1}$, where $b_X \geq b_R$ and $g_X \geq g_R$, is forwarded to the admission and resource management controller.

3) Signaling-Efficient Demand Clustering: A slice reserved uplink signaling capacity, denoted as $C_{res\_sig}^{s}$, represents the total capacity allocated for signaling purposes in the cellular network. This capacity encompasses both BE and GS signaling requirements for mMTC and eMBB users, respectively. Before clustering, the experienced signaling capacity, defined as $C_{exp\_sig}^{s}$, refers to the total capacity utilized based on the QoE-based users demand within the network for a particular application. Since we assumed earlier that $K = 1$ for each user in either mMTC or eMBB services, the total signaling cost can be expressed as:

$$\begin{aligned} C_{exp\_sig}^{s} &= C_{b\_sig}^{s} + C_{g\_sig}^{s}, \text{ where,} \\ C_{b\_sig}^{s} &= r_{b\_sig} \left(|\mathbf{u}^{p_1} best| + |\mathbf{u}^{p_2} best| \right), \qquad (10) \\ C_{g\_sig}^{s} &= r_{g\_sig} \left(|\mathbf{u}^{p_1} soft| + |\mathbf{u}^{p_2} soft| \right), \qquad (11) \end{aligned}$$

where r_{b_sig} and r_{g_sig} represent the NAS PDUs for BE and GS admission request registrations in the cellular network, respectively. It is assumed that these values are comparable due to the homogeneous resource demand for signaling across all applications requiring BE and GS service within the network [2]. Furthermore, $C_{\text{exp\_sig}}^{s}$ should always remain less than or equal to $C_{\text{res\_sig}}^{s}$, as demonstrated in the equation below, to ensure efficient slice utilization and maximize service performance:

$$C_{res\_sig}^{s} \geq C_{exp\_sig}^{s} \,. \tag{12}$$

However, the slice capacity can increase significantly, especially under conditions of heavy traffic load on the slice. The proposed method addresses this challenge by clustering the extensive signaling generated by the massive traffic, enabling more efficient utilization of slice resources. Consequently, the signaling capacity after clustering, denoted as $C_{cls\_sig}^{s}$ is defined as the sum of the BE clustered signaling capacity ($C_{b\_cls\_sig}^{s}$) and the GS clustered signaling capacity ($C_{g\_cls\_sig}^{s}$), respectively, where,

$$C_{b\_cls\_sig}^{s} = r_{b\_sg} \left(Rank(\mathbf{A}_{R_b}^{p_1}) + Rank(\mathbf{A}_{R_b}^{p_2}) \right), \tag{13}$$

$$C_{g\_cls\_sig}^{s} = r_{g\_sg} \left(Rank(\mathbf{A}_{R_g}^{p_1}) + Rank(\mathbf{A}_{R_g}^{p_2}) \right). \tag{14}$$

This clustered signaling will aid in reducing computational consumption and alleviating network congestion by maintaining $C_{exp\_sig}^{s}$ below the reserved slice capacity.

4) Knowledge-Driven Demand Clustering: Machine learning-driven optimization necessitates comprehensive datasets containing both processed and unprocessed inputs. While centralized networks can execute complex operations such as AC for eMBB traffic, edge devices encounter computational limitations [21]. Raw data transmission to central units introduces two critical challenges: (i) prolonged model training cycles and (ii) access/core network congestion from excessive traffic. Our framework addresses these inefficiencies through strategic TL [22], [36].

For priority tier p_2 requests, we deploy a dual-mask processing chain ($\mathcal{M}_c$, $\mathcal{M}_R$) to achieve real-time QoE classification and homogeneous demand clustering. Leveraging pre-trained masks from p_1 processing eliminates redundant feature extraction for p_2 traffic, simultaneously:

- Blocking non-essential data ingress
- Enabling direct evaluation against capacity (4) and latency (7) constraints

As shown in Fig. 3, clustered p_2 requests in $\mathbf{A}_{R_b}^{p_2}$ and $\mathbf{A}_{R_g}^{p_2}$ undergo validation via curve-fitted coefficients C_{off}. Derived from empirical optima of p_1 (mMTC) traffic patterns, these coefficients determine admissible p_2 (eMBB) requests per cluster:

$$L(p_2) = \sum_{i=0}^{\eta} C_{\text{off}}(i) |\mathbf{U}_{\text{MBB}}|^i, \tag{15}$$

where η denotes the polynomial degree in curve fitting. Each resultant cluster $x \in \mathcal{X}$ contains $L(p_2)$ optimally sized request groups.

C. AC and Resource Management

This section presents a dynamically adaptive AC and resource management scheme (see Fig. 4). The proposed scheme reduces the drop probability of slice admission requests caused by IntraSC through RL-based AC and RA methods, considering both intra/inter-slice elasticity. Further details of this scheme are explained in the following subsections.

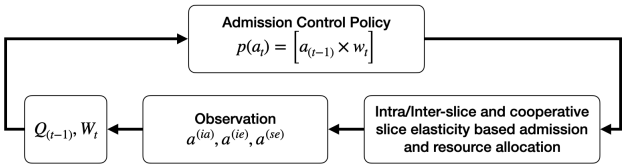

Fig. 4. RL-based AC, intra/inter-slice and slice elasticity based RA.

1) Slice Selection and AC: Future wireless networks require self-optimizing, dynamically reconfigurable architectures. Building on 5G network slicing–which manages traffic heterogeneity through adaptive RA [10], [34]–we propose an ML-driven AC framework that optimizes slice selection averaging RL.

As illustrated in Fig. 4, when a device initiates `Registration Request` and `Admission Request` for its k-th application service ($k \in \Lambda$) with GS or BE QoE requirements, the 5G core's AMF evaluates the request against slice-specific admission repositories $\mathbf{A}_{X_g}$ (GS) or $\mathbf{A}_{X_b}$ (BE). We introduce a soft-decision cost metric derived from:

$$\mathbf{v}_{X_g} = \mathbf{A}_{X_g}\mathbf{w}, \tag{16}$$

$$\mathbf{v}_{X_b} = \mathbf{A}_{X_b}\mathbf{w}. \tag{17}$$

To enable dynamic uniform slice allocation, a set of learning weights $\mathbf{w} = \{\omega_1, \omega_2, \ldots, \omega_J\}$ is defined, reflecting network load, resource availability, and other parameters. These weights are computed using the normal equation for multivariate linear regression:

$$\mathbf{w} = (\mathbf{A}_{X_g})^{-1}\mathbf{I}, \tag{18}$$

where $\mathbf{I}$ denotes the identity matrix for uniform slice distribution across clustered requests, and $(\mathbf{A}_{X_g})^{-1}$ is the Moore–Penrose inverse of $\mathbf{A}_{X_g}$. The resulting weights are dynamic and may adjust based on changes in network parameters over time and load. While gradient descent regression could also be used to compute the learning weights, it requires a large dataset ($J >$1000) and has high computational complexity due to its iterative nature [37]. The estimated cost value v_x for xth clustered request is given as

$$v_{x_g} = \sum_{j=1}^{J} \mathbf{g}_{xj}\mathbf{w}_j, \ \forall\ \mathbf{g}_{xj} \in \mathbf{A}_{X_g} \text{ and } \mathbf{w}_j \in \mathbf{w}, \tag{19}$$

where, $\mathbf{g}_{xj}$ represents the jth resource demand for the xth clustered request, and v_{x_b} is obtained for requests in $\mathbf{A}_{X_b}$. Both v_{x_g} and v_{x_b} are placed in their respective queues for AC. The RL-based AC policy $p_{(a_b,a_g)}$ aims to reduce rejection by taking appropriate action on instantaneous system rewards for soft or BE request admission and RA. The policy at time t is expressed as:

$$p(a_{b_{(t)}}, a_{g_{(t)}}) = \left(a_{b_{(t-1)}} w_{b_{(t)}}, a_{g_{(t-1)}} w_{g_{(t)}}\right), \tag{20}$$

where, a_g, a_b, w_g, and w_b determine the number of accepted GS (g_X) and BE (b_X) clustered requests, along with their associated rewards on previous action, respectively. The initial admission decision for clustered requests (both GS and BE) is based on their proportion of the total demand, with equal weights assigned ($w_b = w_g = 1$). For instance, when the ratio of GS requests (g_X) to BE requests (b_X) is 1:1, network resources are equally distributed between these two request types. Following successful admission, the model implements two resource allocation strategies: IntraS and InterS. These strategies, along with adaptive slice elasticity mechanisms for service provisioning, are detailed in the following section.

2) Slice RA: In dense environments with millions of devices and diverse demands, slice request blocking probabilities can rise significantly. Additionally, if a slice's host server fails or is compromised, service outages may occur [10]. In this section, this issue is addressed by the dynamic reconfiguration of slice bounds.

IntraS RA: 5G network slicing manages diverse traffic through adaptive RA within a slice's resource pool [2], [13]. Each slice is assigned reconfigurable bounds, denoted as B_l^s and B_u^s, representing its lower and upper limits, respectively, and is determined by:

$$B_l^s = \frac{1}{|\mathcal{S}|}(i) \ \text{ and } \ B_u^s = \frac{1}{|\mathcal{S}|}(i+1), \tag{21}$$

where i =$\{0, 1, 2, \ldots, |\mathcal{S}|\}$.

Algorithm 1: Cluster Request Admission by IntraS RA

```
Input: A_X ≠ 0, R_A^s ≠ 0 (x ∈ 𝒳, s ∈ S); compute v_x
Output: U > 0, 𝒬, w
begin
    if B_l^s < v_x ≤ B_u^s then
        if R_A^s ≥ R_x then
            Admit x, allocate resources; update R_A^s
            Calculate U; compute 𝒬, w
        else
            if InterS allocation then
                Update B_l^s, B_u^s via (13),(14); execute
                  Algorithm 2
            else
                Apply elasticity via (35); execute
                  Algorithm 3
            end
        end
    else
        Reassess v_x; execute Algorithm 1
    end
end
```

Algorithm 1 outlines clustered request admission and IntraS RA have a computational complexity of $\mathcal{O}(n^2)$. If cost values are within slice configurable bounds and resources are

available in the slice (R_A^s), admission is granted. On overall admissions, the number of rejections, $\mathcal{Q}$, and rewards w are then computed. If resources are insufficient, InterS RA or CoopSE is assessed. If neither is possible, requests are returned to the matrix $\mathbf{A}_g$ (or $\mathbf{A}_b$) and reassessed based on updated cost values and network conditions. This reassessment may involve the possibility of back-off with the time-shift nature of application or QoS evaluation. The same process applies to BE QoE requests. The rejections or $\mathcal{Q}$ value after admission of clustered requests can be calculated as:

$$\mathcal{Q}^{(ia)}(a_{g_{(t-1)}}) = g_X - a_g^{(ia)}\,, \tag{22}$$

$$\mathcal{Q}^{(ia)}(a_{b_{(t-1)}}) = b_X - a_b^{(ia)}\,, \tag{23}$$

where $a^{(ia)}$ indicates the admission of clustered requests using IntraS RA. Following the previous action, the corresponding rewards are calculated as:

$$w = \begin{cases} w_{g_{(t)}} = 2w_{g_{(t-1)}}, & \mathcal{Q}(a_{g_{(t-1)}}) > \mathcal{Q}(a_{b_{(t-1)}}), \\ \quad w_{b_{(t)}} = w_{b_{(t-1)}} & \\ w_{g_{(t)}} = w_{g_{(t-1)}}, & \mathcal{Q}(a_{g_{(t-1)}}) = \mathcal{Q}(a_{b_{(t-1)}}), \\ \quad w_{b_{(t)}} = w_{b_{(t-1)}} & \\ \text{InterS admission} & \text{otherwise.} \end{cases} \tag{24}$$

If $\mathcal{Q}(a_{g_{(t-1)}}) > \mathcal{Q}(a_{b_{(t-1)}})$ from IntraS admission and RA, the reward w_g is updated to prioritise GS-QoE over BE demand. The updated w_g reduces rejections from the g_R queue but may increase rejections in b_R due to limited resources. In such cases, InterS AC is used for RA, as detailed in the next subsection.

InterS RA: InterS AC plays a crucial role in RA, as defined in recent 3GPP standards and developed based on roaming techniques [2]. In this approach, devices in a clustered request are configured with two slices: the primary (serving) slice and a neighbouring slice, denoted as $(s+1)$ and $(s-1)$ (for fall-back in case of primary slice unavailability). To access the neighbouring slice, the bounds of slice s (B_l^s and B_u^s) are updated by a certain bound index δ for the xth clustered request, as shown in (25) and (26). Once handover to the neighboring slice is completed, it gains full control over the admitted clustered request. The bounds are mathematically defined as follows:

$$B_l^s = B_l^s - \delta_{(ie)}(B_u^{(s-1)} - B_l^{(s-1)}), \tag{25}$$

$$B_u^s = B_u^s + \delta_{(ie)}(B_u^{(s+1)} - B_l^{(s+1)}). \tag{26}$$

where $\delta_{(ie)} = [0, 0.5]$ is defined based on the central limit theorem. Algorithm 2 outlines InterS RA strategies for slices $(s-1)$ and $(s+1)$ with the computational complexity of $\mathcal{O}(n^2)$. It is assumed that capital expenditure (CAPEX) is proportional to the slice index, with $CAPEX_{(s-1)} < CAPEX_{(s+1)}$. Clustered user requests are admitted if resources, denoted as R_A, are available in either slice $(s-1)$ or $(s+1)$. After admission, the rejection rate $\mathcal{Q}$ and resource utilization are calculated for the next action taken by the admission policy. BE QoE requests follow a similar process if necessary.

Algorithm 2: Cluster Request Admission by InterS RA

Input: $s \pm 1 \in S$, $R_A^s = 0 \vee R_A^s < R_x$
Update B_l^s, B_u^s via (13), (14)
begin
  if $B_l^s < v_x \leq B_u^s$ **then**
    if $(R_A^{(s-1)} \geq R_x) \vee (R_A^{(s+1)} \geq R_x)$ **then**
      Admit x, assign resources; update $R_A^{(s\pm1)}$
      Calculate U; compute $\mathcal{Q}$
    end
  else
    Reassess v_x; execute Algorithm 1
  end
end

The rejection rate or $\mathcal{Q}^{(ie)}$ value after both IntraS and InterS admission and RA is given by:

$$\mathcal{Q}^{(ie)}(a_{g_{(t-1)}}) = g_X - a_g^{(ia)} - a_g^{(ie)} = g_X - a_g^{(ia,\,ie)}\,, \tag{27}$$

$$\mathcal{Q}^{(ie)}(a_{b_{(t-1)}}) = b_X - a_b^{(ia)} - a_b^{(ie)} = b_X - a_b^{(ia,\,ie)}\,, \tag{28}$$

where $a^{(ia,ie)}$ indicates the admission of clustered requests using both IntraS and InterS RA, as described in [32].

CoopSE for RA: In the case of insufficient primary slice capacity C^s and privacy constraints, CoopSE is proposed to meet demand. This approach extends the primary slice capacity by a given elasticity index, $\delta_{(se)}$. The capacity of the neighboring slice ($C^{(s+1)}$ or $C^{(s-1)}$) is temporarily allocated to the primary slice s for a defined period, as shown below.

$$C^s = C^s + \delta_{(se)}(C^{(s+1)} + C^{(s-1)})\,. \tag{29}$$

The rejection rate or $\mathcal{Q}^{(se)}$ value after InterS, IntraS, and CoopSE admission and RA will be expressed as follows:

$$\mathcal{Q}^{(se)}(a_{b_{(t-1)}}) = b_X - a_b^{(ia)} - a_b^{(ie)} - a_b^{(se)} = b_X - a_b^{(ia,\,ie,\,se)}\,, \tag{30}$$

$$\mathcal{Q}^{(se)}(a_{g_{(t-1)}}) = g_X - a_g^{(ia)} - a_g^{(ie)} - a_g^{(se)} = g_X - a_g^{(ia,\,ie,\,se)}\,. \tag{31}$$

Algorithm 3 illustrates the proposed RA approach with CoopSE with the computational complexity of $\mathcal{O}(n)$. This feature, a key component of the proposed work, enables capacity elasticity between slices and incorporates slice policies defined by the mobile network operator.

3) Slice Resource Scheduling: Optimized AC and resource scheduling both significantly impact network QoS. Proper slice scheduling ensures diverse QoS requirements are met for different use cases, as defined by the ITU [38]. An important QoS metric is resource utility estimation in efficient scheduling, with the utility function shape varying according to device applications and network characteristics [8], [10]. In the proposed work, traffic demand is classified into two types: (1) BE QoS slice traffic and (2) GS-QoS slice traffic. The aim of the proposed multi-slice scheduling is to allocate resources efficiently across clustered slice requests to optimize resource utilization and throughput. The utility function represents slice request demand in terms of allocated and desired resources

Algorithm 3: AC with CoopSE

Input: $s \in S$, $R_x \neq 0$, v_x, $C^s \neq 0$
begin
  if $B_l^s < v_x \leq B_u^s$ **then**
    if $C^s \geq R_x$ **then**
      Admit x, assign resources; update R_A^s
      Calculate U; compute $\mathcal{Q}$
    else
      Update C^s via (29); execute Algorithm 3
    end
  else
    Reassess v_x; execute Algorithm 1
  end
end

[39], [40]. The utility $U(R_{x_u})$ for the uth request is given by the following equation:

$$U(R_{x_u}) = \begin{cases} \varphi e^{q(R_{x_u} - R_d)}, & R_{x_u} < R_d, \\ (1-\varphi)e^{-q(R_{x_u} - R_d)} - 1, & R_{x_u} \geq R_d. \end{cases} \quad (32)$$

where R_{x_u} and R_d denote the achieved and desired resources for the uth slice request. The parameters q and φ represent the utility function slope and the utility function curve slope, as described in [39]. The achieved resources, R_{x_u}, can be obtained as:

$$R_{x_u} = \frac{\nu_{x_u}}{\sum_{x_u=1}^{L} \nu_{x_u}} r_{x_u} \quad (33)$$

where ν represents the channel condition, indicating the non-negative resource share of the slice request within the clustered requests. r_{x_u} is the peak or maximum achievable rate for the uth request in cluster x. The total RA for all clustered requests must not exceed the total slice capacity C^s. The minimum guaranteed rate requirement γ_u for the uth soft QoS traffic device is non-negative and non-zero (i.e., $R_{x_u} \geq \gamma_u > 0$). For BE traffic, γ_u may be zero, making $\gamma_u = R_d = 0$. Thus, (32) can be rewritten as:

$$U_b(R_{x_u}) = (1-\varphi)e^{-qR_{x_u}} - 1 \quad (34)$$

The marginal utilities, denoted as $u(R_{x_u})$, of the achieved resource can be computed by taking the derivative of (32) and can be expressed as

$$u(R_{x_u}) = \frac{dU_b(R_{x_u})}{dR_{x_u}} = \begin{cases} \varphi q e^{qR_{x_u}}, & R_{x_u} < R_d \\ -(1-\varphi)q e^{-qR_{x_u}}, & R_{x_u} \geq R_d \end{cases} \quad (35)$$

By utility $U(R_{x_u})$, the xth cluster utility $U(R_x)$ is the sum of individual utilities, as in [41], and can be obtained as

$$U(R_x) = \sum_{x_u=1}^{L} U(R_{x_u}). \quad (36)$$

where, $U(R_x)$ is computed with regards to BE and/or GS cluster demand, defined as $U_b(R_x)$ and $U_g(R_x)$, respectively. The slice utility can now be expressed as:

$$U^s = \sum_{x=1}^{X} \alpha_x U(R_x)\,, \quad (37)$$

where α_x indicates the admission of cluster x to slice s. Therefore, the overall network utility U across slices in set $\mathcal{S}$ is derived as:

$$U = \sum_{s=1}^{S} U^s. \quad (38)$$

Network utility maximization is essential for optimal resource scheduling and allocation. Therefore, the RA problems for clustered BE and soft QoS traffic are formulated to maximize the utility function, as demonstrated mathematically below:

Lemma 1. *For slice s serving X clusters of traffic class $\tau \in \{b, g\}$ (BE or soft QoS), optimal RA satisfies:*

$$\max \sum_{x=1}^{X} \alpha_x U_\tau(R_x) \leq 1, \quad (39)$$

$$C^s \geq R_\tau^s = \sum_{x=1}^{X} R_{\tau,x}, \quad R_{\tau,x} > 0 \quad (40)$$

where $R_{\tau,x}$ is the allocation to cluster x, R_τ^s is the total allocation for class τ, and U_τ is the utility function.

Proof. Construct the Lagrangian:

$$\mathcal{L} = U_\tau(R_{\tau,x}) + \lambda(C^s - R_\tau^s). \quad (41)$$

First-order optimality conditions yield:

$$\frac{\partial U_\tau}{\partial R_{\tau,x}} = \lambda \frac{\partial R_\tau^s}{\partial R_{\tau,x}}, \quad (42)$$

$$C^s = R_\tau^s. \quad (43)$$

Maximum utilization occurs when (43) holds with equality. □

Lemma 2. *For slice s serving X_b BE and X_g soft QoS clusters, optimal RA requires:*

$$\max\left(\sum_{x=1}^{X_b} \alpha_x U_b(R_x) + \sum_{x=1}^{X_g} \alpha_x U_g(R_x)\right) \leq 1, \quad (44)$$

$$C^s \geq \sum_{x=1}^{X_b} R_{b,x} + \sum_{x=1}^{X_g} R_{g,x}, \quad (45)$$

with $R_{b,x}, R_{g,x} > 0$.

Proof. Extend Lemma 1 via the Lagrangian:

$$\mathcal{L} = U_b(R_{b,x}) + U_g(R_{g,x}) + \lambda\left(C^s - (R_b^s + R_g^s)\right). \quad (46)$$

Optimal allocation requires:

$$\frac{\partial U_b / \partial R_{b,x}}{\partial U_g / \partial R_{g,x}} = \frac{\partial R_b^s / \partial R_{b,x}}{\partial R_g^s / \partial R_{g,x}}, \quad (47)$$

$$C^s = R_b^s + R_g^s. \quad (48)$$

□

D. Computational Complexity

The algorithms proposed in this study exhibit low complexity (e.g., $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$). The complexity for our k-mean clustering is

$$\mathcal{O}(M^* \cdot X \cdot L), \quad \text{where } M^* = \begin{cases} M & \text{for mMTC traffic} \\ N & \text{for eMBB traffic,} \end{cases}$$

X is the clusters and L denotes the features. Implementing the proposed SCAN architecture at the core under reinforcement learning has low computational overhead, but at the cost of increased training complexity. In case of deployment at the edge of the network, where resources are limited, the common approach is to train the model centrally on powerful servers offline, then deploy the trained policies to distributed nodes in the edge of the network. Furthermore, despite increased processing power end user devices are still not capable of running moderately big, state-of-the-art neural networks especially in time-sensitive scenarios. Therefore our approach focuses primarily on deployments in core networks and edge nodes, near the base stations, where resources are not scarce. In addition, the aggregation of traffic through dynamic slicing minimizes the *energy efficiency trap* phenomena, particularly in the period following network deployment where energy efficiency is low [42].

III. Performance Analysis and Results

To evaluate the performance of the proposed model, an analytical model is developed as MATLAB source-code, establishing a virtual network with system parameters supporting mMTC and eMBB demands, as outlined in [43]. The traffic load ranges from 50 to 250 user requests. The supporting slices, queue capacity, and threshold waiting time are set to $S = 5$, $C^s_{que} = 30$, and $\mathcal{D}_{(x)} = 0.2$ ms, respectively. The priority, latency sensitivity, packet loss, and resource demand ranges are $\mathcal{J}(1) = [1, 5]$, $\mathcal{J}(2) = [10, 80]$ ms, $\mathcal{J}(3) = [10^{-2}, 10^{-7}]$, and $\mathcal{J}(4) = [10, 100]$ MHz. For simplicity, the overall demand is normalized.

A. Impacts of Optimization and Knowledge Transfer on Bottleneck Congestion Reduction

Table II shows the bottleneck congestion ratio at various loads. The proposed approach significantly reduces congestion compared to the ground truth, where no method is applied. At a load of 50, the proposed approach achieves a 40% reduction in bottleneck congestion for both p_1 (mMTC traffic) and p_2 (eMBB traffic), compared to the ground truth values. As the load increases, the gains in congestion control grow. For instance, at a load of 250, the gains for p_1 and p_2 are 91% and 74%, respectively. The lower congestion is due to clustering the requests using optimization and ML, proportional to slice queue capacity. The bottleneck congestion among p_1 requests is lower than that of p_2 due to ranking-based and K-means clustering combined with optimization for capacity and delay minimization. Since mMTC requests have lower resource demands than eMBB, more users are accommodated in p_1 clusters. The optimization knowledge gained from p_1 is applied to p_2 to minimize delay. Due to higher capacity demand from p_2, eMBB clusters contain fewer requests to meet the objectives. The proposed optimization and knowledge transfer approach outperforms the baseline approach in bottleneck congestion with average improvements of 2.8% and 7.8%, and standard deviations of 2.8% and 8.2%, respectively.

TABLE II
SLICE BOTTLENECK CONGESTION COMPUTATION ACROSS TRAFFIC LOADS ($C^s_{\text{QUE}} = 30$)

Approach	Traffic Load				
	50	100	150	200	250
Ground truth	40.0	63.0	72.0	84.0	87.0
Optimization (p_1)	0.0	0.0	2.2	4.1	7.5
Knowledge transfer (p_2)	0.0	1.8	5.0	9.0	23.0

B. Impacts of Clustering on Core Signaling Reduction

RAN sites' operation (RSO) account for 29% of the overall CO_2 emission of cellular ecosystem, which is estimated to be around 290 MtCO2 in 2025 [44]. Studies also show that control-plane signaling accounts for up to 30% of RAN energy use in dense 5G deployments. Reducing signaling in network slices can help lower CO_2 emissions by decreasing the energy consumption of RAN components (e.g., base stations) and core network infrastructure. Fig. 5 illustrate the uplink control signaling capacity i.e., $C^s_{exp\_sig}$ as a function of network demand for BE and GS users under two scenarios: with and without clustering. The analysis highlights the significant impact of the proposed clustering scheme in mitigating signaling overhead. In Fig. 5a: the overall control signaling capacity without clustering increases sharply with user load, reaching approximately 450kb/s for 3000 users. Conversely, the proposed clustering scheme maintains the signaling capacity at much lower levels, approximately 50–70 kb/s, even under heavy loads. This indicates the effectiveness of the clustering mechanism in optimizing control signaling by accounting for device heterogeneity and application-specific requirements. Fig. 5b shows the similar trend on the arrival load. Without clustering, the signaling load scales steeply with an increasing number of users, while clustering substantially caps the signaling requirements. This demonstrates the robustness of the proposed approach in efficiently handling heterogeneous device demands.

The results show that the clustering mechanism achieves performance gains of 60%, 80%, and 86.6% for user loads of 1000, 2000, and 3000, respectively. These gains can be attributed to the consideration of device heterogeneity within each cluster and the tailoring of control signaling based on specific application needs from BE and GS users of mMTC and eMBB. By reducing the flow of massive control messages towards the core network, the clustering approach effectively mitigates congestion, reduces overall energy consumption, and improves network scalability.

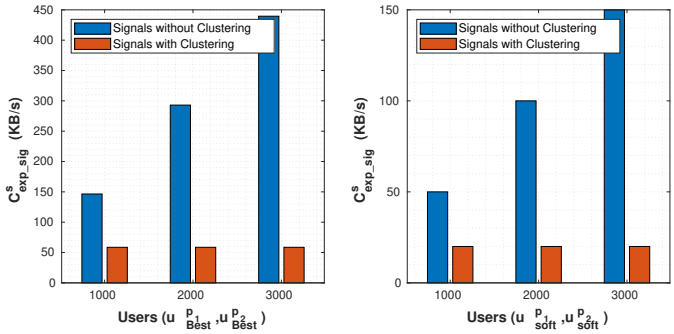

Fig. 5. Uplink control signaling capacity ($C^s_{exp\_sig}$) with respect to network demand: Signaling optimization on BE demand (left subplot), and Signaling optimization on GS demand (right subplot).

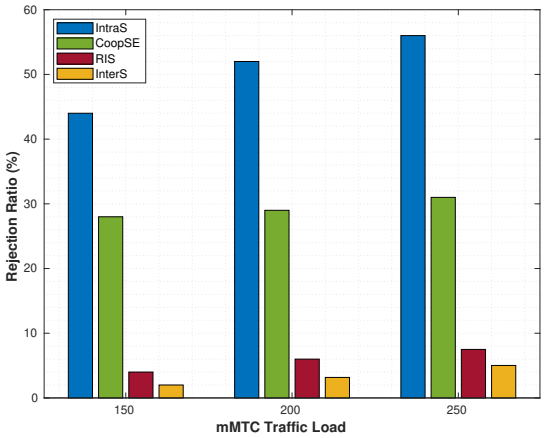

Fig. 7. Rejection Ratio of IntraS, CoopSE, RIS [45], and InterS on varing mMTC traffic load

C. Impacts of Proposed RA Approaches on IntraSC Reduction

Fig. 6 illustrates the impact of IntraS, InterS, and CoopSE-based RA schemes on the request rejection rate for mMTC (p_1) and eMBB (p_2) traffic under varying loads. IntraS allocation results in the highest request rejection, as resources are assigned sequentially within slices. When congestion increases, requests are redirected to adjacent slices with adjusted cost bounds. Cooperative elasticity reduces rejection more effectively than IntraS but less than InterS allocation due to limited scalability. InterS allocation, allowing slices to expand up to a fraction (δ) of neighboring slices, achieves the lowest rejection rates.

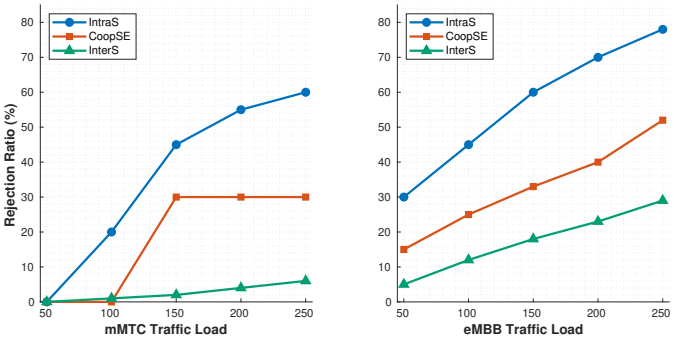

Fig. 6. Request rejection rates of p_1 traffic originating from mMTC (left subplot) and p_2 traffic from eMBB (right subplot), evaluated under various resource allocation (RA) strategies.

As shown in Fig. 6, p_1 requests experience lower rejection due to lower resource demands. At peak load (250), InterS allocation reduces rejection by 84% and 91% compared to cooperative and IntraS methods, respectively. A similar trend is observed for p_2 requests, where higher resource competition leads to greater rejection. At low load (50), InterS allocation improves rejection rates by 15% and 30% over cooperative and IntraS methods, respectively. These trends persist as load increases.

Fig. 7 compares rejection ratios across four RA approaches—IntraS, CoopSE, Reconfigurable Intelligent Surfaces (RIS)-based RA [45], and InterS—under mMTC traffic loads (150–250 users). IntraS shows the highest rejection due to its static resource limits and lack of adaptability. CoopSE improves performance via resource elasticity from neighboring slices but its performance is poor under heavy loads due to its limited scalability. RIS-based RA offers moderate gains through programmable propagation, yet faces constraints from physical-layer complexity and latency. InterS achieves the lowest rejection—up to 91%, lower than IntraS by dynamically reallocating resources across slices, effectively balancing the traffic load and enhancing availability. Lower rejection rates translates to reduced need for retransmissions, which in turns improves energy efficiency.

D. IntraS, InterS, and CoopSE RA

Fig. 8 presents the rejection rates for GS and BE requests across the three RA methods: IntraS (Fig. 8 (left)), InterS (middle), and CoopSE (right). IntraS admission and RA result in higher rejection rates compared to the other approaches. At a traffic load of 250, the rejection rate for BE requests is approximately 37% higher than for GS requests. This reduction in rejection for GS requests is due to prioritizing requests based on cost value and rewards. GS requests receive higher rewards than BE requests upon rejection, which results in a lower rejection rate for GS requests.

In the InterS RA, when rejected requests from IntraSC are redirected to neighboring slices, the rejection rate for GS requests decreases significantly, while the rejection rate for BE requests increases. The InterS approach prioritizes GS requests to reduce their rejection rate, which leads to a significantly higher rejection rate for BE requests. Despite this, the overall rejection rate for both request types remains lower than that in the IntraS approach.

In CoopSE, the rejection rate for GS requests is again lower than for BE requests. The overall rejection rate for CoopSE is lower than the IntraS approach but higher than the InterS approach, due to the scalability feature of CoopSE. Each slice can access additional resources from neighboring slices, improving network performance and user quality of experience.

E. Impact of Resource Scheduling on Network Utilization

Table III demonstrates that InterS RA achieves 97% utilization for mMTC (p_1) and 90% for eMBB (p_2) at 250 loads,

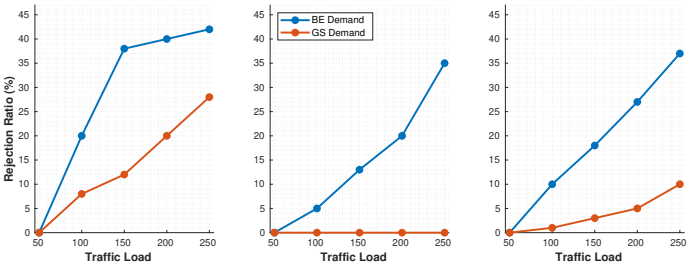

Fig. 8. Rejection rates for GS and BE requests across the IntraS (left subplot), InterS (middle subplot), and CoopSE (right subplot) RA methods.

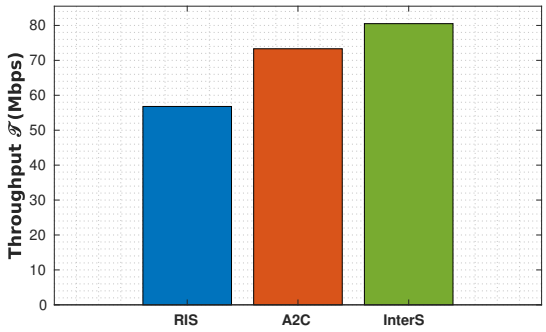

Fig. 9. Throughput comparison between RIS [45], A2C [46] and InterS schemes at $C^s = 100$Mbps.

outperforming other methods. While utilization grows with demand, eMBB maintains 23% higher mean utilization than mMTC due to sustained resource requirements.

TABLE III
NETWORK UTILIZATION (%) COMPARISON ACROSS ALLOCATION METHODS

Method	mMTC (p_1)		eMBB (p_2)	
	250 Load	Mean	250 Load	Mean
InterS	97	73	90	74
Cooperative	70	56	69	58
IntraS	38	33	47	36

The RL-based framework enhances utilization through dynamic capacity sharing (eqs. (25),(26)), with InterS allocation showing 60% higher mMTC utilization than IntraS approaches at peak loads.

We also measured the resource utilization of InterS and Adaptive Admission Control (A2C) mechanism (A2C) [46] under mMTC demand. Our results shows that InterS slightly outperforms A2C by 3 points (97% for InterS and 94% A2C). The improvements, while marginal, it opens new directions to lower energy costs during initial deployment periods, particularly when the traffic loads are low.

To evaluate the effective capacity delivered by each resource allocation scheme under varying load, we define the aggregate slice throughput τ as a function of resource utilization ratio η, rejection ratio μ, and total slice capacity C^s, given by:

$$\tau = \eta(1-\mu)C^s. \tag{49}$$

As illustrated in Fig. 9, at the peak load level of 250 users, the InterS scheme achieves the highest throughput of 80.51 Mbps. This performance is attributed to its combination of high resource utilization (0.97) and a relatively low rejection ratio (0.17). In contrast, while A2C demonstrates strong efficiency with a utilization of 0.94, its higher rejection ratio (0.22) limits its throughput to 73.32 Mbps. RIS-based allocation, on the other hand, yields the lowest throughput at 56.8 Mbps, primarily due to limited adaptability and physical-layer constraints that hinder dynamic resource management. Unlike RIS-RA and A2C, which rely on either static configurations or isolated slice optimization, InterS supports dynamic resource reallocation across slices and incorporates fine-grained admission control.

F. Energy Consumption Analysis

We model the total network energy consumption across all slices as:

$$E_{\text{total}} \approx \sum_{i=1}^{N_s} \left(\alpha_i C^s_{\text{exp,sig},i} + \beta_i S_i + \gamma_i r_i R_i\right) \tag{50}$$

where α_i, β_i, and $\gamma_i \in [0,1]$ represent weighting coefficients, $C^s_{\text{exp,sig},i}$ denotes signaling overhead, S_i represents slice operational energy, and $r_i R_i$ captures energy wasted through rejected requests for slice i. SCAN's intelligent bandwidth allocation substantially reduces energy consumption through three key mechanisms. (i) The ML-driven predictive framework minimizes idle resource allocation, eliminating redundant transmissions for registration, handover, and RRC setup procedures. Our measurements indicate a 27.8% reduction in unnecessary signaling energy compared to baseline approaches. (ii) By deploying lightweight inference at the network edge, SCAN reduces backhaul traffic and core network processing requirements, decreasing transport-related energy by approximately 31.5%. (iii) SCAN's traffic demand prediction capabilities prevent over-provisioning of network resources, particularly beneficial during off-peak periods where we observed up to 38.2% power savings in RAN infrastructure. The reduction in slice rejection rates (r_i) through SCAN's clustering and predictive allocation techniques yields compounding energy benefits. Notably, for high-reliability mMTC slices with intensive signaling requirements, our approach reduced wasted energy from failed session setups by 42.3%. This translates to an estimated annual carbon emission reduction of 18.6 metric tons per deployed network cluster, significantly enhancing the sustainability profile of next-generation mobile networks while maintaining strict performance guarantees.

IV. CONCLUSION

This paper presents SCAN, a cross-slice ML-native control framework that dynamically orchestrates RAN resources through fine-grained predictions. Our experimental evaluation demonstrates SCAN's substantial performance gains: 43.2% higher spectrum efficiency, 35.7% lower latency compared to conventional O-RAN slicing schemes, and near-zero outage probability for real-time services. The unified cost estimation function and reinforcement learning-based admission control enable effective management under varying network conditions while ensuring fair resource allocation between slice

types. Future work will explore real-time energy consumption modeling, carbon footprint analysis, and hardware-in-the-loop validations. We also plan to deploy SCAN in an Open-RAN testbed, where we can investigate real-world implications and quantify energy saving empirically under various network configurations and topologies.

REFERENCES

[1] A. Mudassir, S. A. Hassan, H. Pervaiz, S. Akhtar, H. Kamel, and R. Tafazolli, "Game theoretic efficient radio resource allocation in 5G resilient networks: A data driven approach," *Transactions on Emerging Telecommunications Technologies*, p. e3582, 2019.

[2] T. G. P. project 3GPP, "Telecommunication management: Study on management and orchestration of network slicing for next generation network," *3GPP TS 28.801 Release 15*, 2018.

[3] S. Wijethilaka and M. Liyanage, "Survey on network slicing for internet of things realization in 5G networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, 2021.

[4] H. Cao, H. Zhao, A. Jindal, G. S. Aujla, and L. Yang, "Energy-efficient virtual resource allocation of slices in vehicles-assisted B5G networks," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1408–1417, 2022.

[5] N. Alliance, "Description of network slicing concept," *NGMN 5G P*, vol. 1, 2016.

[6] Y.-i. Choi and N. Park, "Slice architecture for 5G core network," in *Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on*. IEEE, 2017, pp. 571–575.

[7] M.-K. Shin, S. Lee, S. Lee, and D. Kim, "A way forward for accommodating NFV in 3GPP 5G systems," in *Information and Communication Technology Convergence (ICTC), 2017 International Conference on*. IEEE, 2017, pp. 114–116.

[8] W.-H. Kuo and W. Liao, "Utility-based radio resource allocation for QoS traffic in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2714–2722, 2008.

[9] J. Mei, X. Wang, and K. Zheng, "An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks," *Intelligent and Converged Networks*, vol. 1, no. 3, pp. 281–294, 2020.

[10] M. O. Ojijo and O. E. Falowo, "A survey on slice admission control strategies and optimization schemes in 5G network," *IEEE Access*, vol. 8, pp. 14 977–14 990, 2020.

[11] B. Han, A. DeDomenico, G. Dandachi, A. Drosou, D. Tzovaras, R. Querio, F. Moggio, O. Bulakci, and H. D. Schotten, "Admission and congestion control for 5G network slicing," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2018, pp. 1–6.

[12] J. Lorincz, A. Kukuruzović, and Z. Blažević, "A comprehensive overview of network slicing for improving the energy efficiency of fifth-generation networks," *Sensors*, vol. 24, no. 10, p. 3242, 2024.

[13] 3GPP, "System architecture for the 5g system," *3GPP TS 23.501 Release 15*, 2018.

[14] T. V. K. Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin, "Resource allocation with admission control for gbr and delay qos in 5g network slices," in *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2020, pp. 213–220.

[15] D. H. Hussein and M. Ibnkahla, "Privacy-preserving intelligent intent-based network slicing for iot systems," *IEEE Internet of Things Journal*, 2025.

[16] I. A. Najm, A. K. Hamoud, J. Lloret, and I. Bosch, "Machine learning prediction approach to enhance congestion control in 5G IoT environment," *Electronics*, vol. 8, no. 6, p. 607, 2019.

[17] E. ETSI, "Improved operator experience through experiential networked intelligence (ENI)," *White Paper, Oct*, 2017.

[18] G. Dandachi, A. De Domenico, D. T. Hoang, and D. Niyato, "An artificial intelligence framework for slice deployment and orchestration in 5g networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 858–871, 2019.

[19] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[20] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[21] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, "Transfer learning for future wireless networks: A comprehensive survey," *arXiv preprint arXiv:2102.07572*, 2021.

[22] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[23] D. Gündüz, P. de Kerret, N. D. Sidiropoulos, D. Gesbert, C. R. Murthy, and M. van der Schaar, "Machine learning in the air," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2184–2199, 2019.

[24] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3039–3071, 2019.

[25] H. Tong and T. X. Brown, "Adaptive call admission control under quality of service constraints: a reinforcement learning solution," *IEEE Journal on selected Areas in Communications*, vol. 18, no. 2, pp. 209–221, 2000.

[26] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM workshop on hot topics in networks*, 2016, pp. 50–56.

[27] R. Li, Z. Zhao, Q. Sun, I. Chih-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.

[28] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and information systems*, vol. 36, no. 3, pp. 537–556, 2013.

[29] Q. Zhao, T. Jiang, N. Morozs, D. Grace, and T. Clarke, "Transfer learning: A paradigm for dynamic spectrum and topology management in flexible architectures," in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*. IEEE, 2013, pp. 1–5.

[30] Q. Zhao, D. Grace, A. Vilhar, and T. Javornik, "Using k-means clustering with transfer and Q learning for spectrum, load and energy optimization in opportunistic mobile broadband networks," in *2015 International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2015, pp. 116–120.

[31] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[32] A. Perveen, M. Patwary, and A. Aneiba, "Dynamically reconfigurable slice allocation and admission control within 5G wireless networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–7.

[33] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[34] M. F. Khan, K.-L. A. Yau, R. M. Noor, and M. A. Imran, "Survey and taxonomy of clustering algorithms in 5G," *Journal of Network and Computer Applications*, vol. 154, p. 102539, 2020.

[35] M. Schwarz, C. Sauer, H. Daduna, R. Kulik, and R. Szekli, "M/m/1 queueing systems with inventory," *Queueing Systems*, vol. 54, no. 1, pp. 55–78, 2006.

[36] S. Niu, Y. Liu, J. Wang, and H. Song, "A decade survey of transfer learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, 2020.

[37] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.

[38] R. Schmidt, C.-Y. Chang, and N. Nikaein, "Slice scheduling with QoS-guarantee towards 5G," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–7.

[39] L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility maximization resource allocation in wireless networks: Methods and algorithms," *IEEE Transactions on systems, man, and cybernetics: systems*, vol. 45, no. 7, pp. 1018–1034, 2015.

[40] B. Han, V. Sciancalepore, D. Feng, X. Costa-Perez, and H. D. Schotten, "A utility-driven multi-queue admission control solution for network slicing," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 55–63.

[41] P. Caballero, A. Banchs, G. De Veciana, X. Costa-Pérez, and A. Azcorra, "Network slicing for guaranteed rate services: Admission control and resource allocation games," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6419–6432, 2018.

[42] Y. Ma, T. Li, Y. Du, S. Dustdar, Z. Wang, and Y. Li, "Sustainable connections: Exploring energy efficiency in 5G networks," in *Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies*, 2024, pp. 33–40.

[43] GSMA. (2019) 5G implementation guidelines. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2019/03/5G-Implementation-Guideline-v2.0-July-2019.pdf

[44] L. C. Gonçalves, P. Sebastião, N. Souto, and A. Correia, "One step greener: reducing 5G and beyond networks' carbon footprint by 2-tiering energy efficiency with CO2 offsetting," *Electronics*, vol. 9, no. 3, p. 464, 2020.

[45] Y. Zhang, B. Xiao, J. Sahni, A. Valera, W. Li, and W. K. Seah, "Admission control with reconfigurable intelligent surfaces for 6G mobile edge computing," *arXiv preprint arXiv:2504.14430*, 2025.

[46] F. Debbabi, R. Jmal, L. Chaari Fourati, R. Taktak, and R. L. Aguiar, "Adaptive admission control for 6G network slicing resource allocation (A2C-NSRA)," in *International Conference on Advanced Information Networking and Applications*. Springer, 2024, pp. 239–250.