Semi-Contention-Free Access in IoT NOMA Networks: A Reinforcement Learning Framework

Abhishek Kumar, José-Ramón Vidal, Jorge Martinez-Bauset, and Frank Y. Li

***Abstract*—The unprecedented surge of massive Internet of things (mIoT) traffic in beyond fifth generation (B5G) communication systems calls for transformative approaches for multiple access and data transmission. While classical model-based tools have been proven to be powerful and precise, an imminent trend for resource management in B5G networks is promoting solutions towards data-driven design. Considering an IoT network with devices spread in clusters covered by a base station, we present in this paper a novel model-free multiple access and data transmission framework empowered by reinforcement learning, designed for power-domain non-orthogonal multiple access networks to facilitate uplink traffic of small data packets. The framework supports two access modes referred to as contention-based and semi-contention-free, with its core component being a policy gradient algorithm executed at the base station. The base station performs access control and optimal radio resource allocation by periodically broadcasting two control parameters to each cluster of devices that considerably reduce data detection failures with a minimum computation requirement on devices. Numerical results, in terms of system and cluster throughput, throughput fairness, access delay, and energy consumption, demonstrate the efficiency and scalability of the framework as network size and traffic load vary.**

***Index Terms*—Massive Internet of things, uplink small data packet, semi-contention-free and contention-based, reinforcement learning and policy gradient, performance evaluation.**

I. Introduction

Along with the intensive deployment of fifth generation (5G) mobile networks worldwide focusing on enhanced mobile broadband (eMBB) services, ubiquitous connectivity in the form of massive Internet of things (mIoT) heralds a transformative evolution towards sixth generation (6G) [1]. According to ITU [2], the research target of connection density could reach $10^6 \sim 10^8$ devices per square kilometer in 2030.

Although a huge amount of IoT devices may be deployed across a cell, most of them are battery-powered devices and require typically infrequent data reporting or data collection [3]. Indeed, IoT traffic exhibits a characteristic of *small data* in terms of short packet size and sporadic data transmissions. In the presence of overlapping transmissions based on the same radio resource, network performance may be degraded due to the reception of multiple signals that make the correct detection of individual signals impossible. Among various mechanisms to diminish data detection failures, non-orthogonal multiple access (NOMA) and clustering are probably the two most eminent techniques. Through power-domain NOMA, concurrent transmissions from multiple users on the same radio resource can be disentangled and one or multiple transmissions may be successfully detected through successive interference cancellation (SIC). On the other hand, clustering which may be facilitated by multiple-input–multiple-output (MIMO)-based beamforming and beam steering can mitigate interference generated by users from other clusters [4]. By jointly performing NOMA and clustering, the number of devices covered by the same beam could be greatly reduced and the benefit of SIC would be more significant.

However, dealing with a substantial number of simultaneous competing transmissions in clustered NOMA networks is not an easy task as both intra-cluster and inter-cluster interference[1] exists and no exact information on instantaneous access demands for uplink IoT traffic is known to a base station (BS). Furthermore, how to assess the behavior of such networks analytically and to design access schemes to optimize network performance *in a real-time manner* call for further research endeavors, *especially when network size grows*.

To assess the performance of a network, analytical models for example those that are built based on Markov chains have been proven to be a powerful tool, as they lead to closed-form expressions for various performance metrics [5] [6]. However, such analytical solutions may face scalability difficulties with respect to network size and complexity. On the other hand, the recent advances in artificial intelligence (AI) and machine learning (ML) enable numerous *data-driven* approaches that envisage great potential to both providing real-time adaptation to dynamic network conditions and solving the scalability problem. In addition to the surge of interests in academia, the 3rd generation partnership project (3GPP) is also promoting ML applications in 6G radio access networks [7]. Among a catalog of various ML approaches, reinforcement learning (RL) [8] appears as a promising category to address the network scenario envisaged in this study.

In the paper, we study a communication network where IoT devices perform *uplink random access and data transmission* in a hybrid semi-contention-free (SCF) or/and contention-based (CB) manner without requiring pre-allocated radio re-

Manuscript received February 18, 2025; revised June 6, 2025 and August 12, 2025. The research leading to these results has received funding from the European Economic Area (EEA) Norway (NO) Grants 2014-2021, under Project contract no. 42/2021, RO-NO-2019-0499 –"A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services (SOLID-B5G). The work of José-Ramón Vidal and Jorge Martinez-Bauset was also supported by Grant PID2021-123168NB-I00, funded by MCIN/AEI, Spain/10.13039/501100011033 and the European Union A way of making Europe/ERDF. (Corresponding author: Frank Y. Li.)

Abhishek Kumar and Frank Y. Li are with the Department of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway (email: {abhishek.kumar; frank.li}@uia.no).

José-Ramón Vidal and Jorge Martinez-Bauset are with the Departamento de Comunicaciones, Universitat Politècnica de València (UPV), 46022 València, Spain (email: {jrvidal; jmartinez}@upv.es).

[1] While intra-cluster interference indicates the interference generated by concurrent transmissions from other devices in the same cluster, inter-cluster interference is caused by the transmissions from any other cluster(s) [6].

TABLE I: Transposed comparison of learning-based clustering, access, and transmission schemes for uplink traffic in NOMA networks

Category	**Our schemes**	**[18]**	**[19]**	**[20]**	**[21]**	**[23]**
Approach	PG-based model-free access control and transmission for small data	ANN trained on heuristic clustering	Hybrid SARSA and DRL framework using 3D state actions	EM-based probabilistic clustering with online update	Optimization-based power and user allocation	LSTM-based DRL for sub-channel and power level selection
Clustering Strategy	Location-based clustering and probabilistic user pairing	SIC capability-aware user partitioning	Joint user, BS, and sub-channel mapping via state learning	Location-aware Gaussian mixture via unsupervised learning	Clustered access for hybrid spectrum sharing	Device sub-channel clustering to reduce collisions
Learning Algorithm	Reinforcement learning (policy gradient)	Offline supervised ANN	SARSA (light) + DQN (dense)	Statistical model-based (EM + online EM)	Non-learning (convex optimization)	DQN with LSTM
System Type	Uplink mIoT-NOMA	mmWave-NOMA	NOMA-IoT uplink	Downlink mmWave NOMA	Cognitive IoT uplink NOMA	Grant-free NOMA uplink
Key Contributions	RL-driven unified access control for CB and SCF modes; low device complexity	Heuristics for label generation and inference in real-time	Traffic-aware switching between RL types	Fixed & dynamic user clustering; reduced re-clustering complexity	Support of both PU-first and PU-last decoding orders	Learning-based contention patterns for access success
Performance Highlights	Scalable system/ cell throughput; low delay; and efficient energy consumption	Heuristic performance with faster decision latency	Effective handling of dynamic traffic while maintaining fairness	Full EM performance while reducing runtime for dynamic scenarios	Balanced primary user protection and system throughput under hybrid access	Effective access policy and reduced contention over time

sources. From this perspective, the proposed access modes can be considered as a class of *grant-free* (GF) random access. We regard the SCF access mode as an enhanced access mode that substantially improves the performance of conventional CB access by reducing both intra- and inter-cluster interference.

As described latter, an RL-agent located at the BS performs access control and determines radio resource allocation policies, based solely on its observation of the number of successful *small data packet* (SDP) detections in the uplink traffic. As an additional advantage, the proposed SCF access mode requires minimal computation capability on devices. The access schemes proposed in this study resemble the principle of *random access small data transmission* described in [10], however, with much more advanced features facilitated by RL-driven access control and radio resource allocation.

A. Related Work

1) NOMA, clustering, GF/semi-GF (SGF): Among various access and data transmission schemes, many studies considered NOMA and clustering as promising bases for their scheme design. While NOMA can facilitate concurrent transmissions, clustering can further exploit the benefits of NOMA through device pairing [11]. In [12], a priority access NOMA-based slotted ALOHA scheme was proposed, supporting multiple power levels and priority levels. However, their proposed traffic estimator relays on the existence of certain degree of symmetry between uplink and downlink traffic which might not be realistic for IoT traffic.

Considering both intra- and inter-cluster interference in NOMA transmission, joint user clustering and power control for uplink traffic were studied in [13]. Recently, a contention-based coded random access scheme for heterogeneous traffic that allows both time and frequency domain resource sharing was proposed in [14]. Although these two studies investigated important *physical layer aspects*, they did not consider a dynamic traffic scenario where random traffic is generated by devices, nor did they evaluate performance parameters such as throughput, delay, or fairness. Furthermore, we clarify that investigating such physical layer techniques is beyond the scope of this paper.

In [15], an access mode referred to as SGF was proposed, allowing GF devices share the dedicated radio resources allocated to grant-based (GB) devices by exploiting NOMA and SIC. In an additional study, the authors proposed an enhanced SGF scheme to guarantee certain degree of quality of service (QoS) to GB devices [16]. Another recent paper proposed a novel SGF access scheme for short packet transmission by improving device-to-slot allocations based on the partial information observed by a BS [17]. These proposals were developed based on an assumption that either the BS or the GF users can acquire perfect knowledge of the channel state information (CSI). However, such an assumption seems to be unrealistic in real-life scenarios, particularly when a massive number of IoT devices contend for access and when considering the randomness of IoT data traffic [14].

2) Data-driven learning-based access schemes: To design access and transmission schemes and analyze the performance of NOMA-enabled concurrent data transmissions require novel techniques. As a trend beyond various conventional *model-based* methods, there is a surge of endeavors in recent years to explore *data-driven* learning-based approaches including RL-enabled access to uplink data transmissions as well as their potential benefits.

In [18], two artificial neural network (ANN) algorithms that assign users to clusters in millimeter wave-NOMA (mmWave-NOMA) networks in a real-time manner were proposed. In [19], an RL-based resource allocation scheme was developed for uplink data transmissions in clustered NOMA

IoT networks using deep reinforcement learning (DRL) and state–action–reward–state–action (SARSA)-learning algorithms based on three-dimensional (3D) state and action associations. In [20], an expectation maximization (EM)-based online clustering algorithm which is able to update user clusters through unsupervised learning based on location-awareness was developed. Furthermore, [21] proposed a NOMA-based hybrid spectrum access scheme for uplink cognitive IoT traffic, supporting both decoding-primary-user(PU)-last and decoding-PU-first optimization.

In [22], a Q-learning (QL)-based random access method for NOMA IoT networks was proposed. A deep RL algorithm for throughput enhancement in GF NOMA systems was proposed in [23] and two distributed QL algorithms for GF uplink transmissions in the presence of bursty IoT traffic were proposed in [24]. Moreover, a deep RL-based learning access scheme for signature-based GF transmissions was developed in [25], targeting at maximizing long-term successful transmissions in NOMA-enabled beyond 5G (B5G) networks. Similarly, [26] introduced a double deep QL algorithm for efficient resource allocation in NOMA networks that handles dynamic traffic patterns. Another study in [27] explored multi-agent RL techniques for enhancing uplink GF NOMA performance, resulting in improved throughput and fairness. Moreover, [28] presented a novel RL-based power control mechanism combined with GF access to minimize energy consumption and maximize system reliability in clustered IoT networks. In a preliminary phase of this study, we proposed an RL-based random access scheme for IoT traffic where actions for access control were performed through QL [29]. The aforementioned efforts to a certain extent unveil the promising role that RL may play in overcoming the complexities of medium access and data transmissions in B5G networks. In Table I, we provide a transposed comparison of our schemes with five representative schemes related to learning-based clustering, access, and transmissions in NOMA-enabled networks for IoT uplink traffic.

However, a number of research questions need to be answered before an RL-based approach can be applied to real-life B5G/6G networks. Among those, the following four questions triggered our motivation to perform the study reported in this paper. 1) What are the performance benefits that can be obtained by allowing multiple devices that perform concurrent SDP transmissions based on SCF and CB access share uplink radio resources? 2) How can a BS determine high performance access control policies given that it can only observe the number of successful SDP transmissions and it does not know the number of devices simultaneously attempting to access common uplink radio resources? 3) How close is the system performance achieved by the access control policies computed by an RL algorithm fed with partial system information from the ideal system performance? and 4) How does the complexity and performance of an RL-based access scheme scale with network size and traffic load?

B. Contributions

When concurrent transmissions of two or more devices from one or multiple clusters occur on the same radio resource, the intricacies of SIC detection surge, as SIC alone may not resolve a collision when the obtained signal-to-interference-and-noise ratio (SINR) is not large enough [6]. To minimize SDP detection failures, a BS may impose access control to devices in order to limit the number of concurrent access to uplink radio resources.

Among various existing studies on NOMA-enabled IoT data transmissions, both *saturated* and *non-saturated* traffic conditions have been considered [14] [23]. In the saturated case, the BS knows that all devices always have a packet to transmit [23]. In the non-saturated case, where traffic state is unknown to the BS, packet arrival *prediction* or *estimation* serves as the basis for the design of an access control scheme [5]. However, such prediction schemes often rely on assumptions like Poisson arrivals or traffic models that may not hold in reality.

In this paper, we introduce a random access and SDP transmission framework for uplink IoT traffic, referred to as RL for semi-contention-free (RL4SCF), and propose a *data-driven model-free* RL-enabled access control and radio resource allocation mechanism that supports both the SCF and the CB random access modes. The operation of RL4SCF is *observation-based*, without the need to assume any specific patterns of packet arrivals, nor is it necessary for the BS to know the instantaneous traffic load within the cell. In addition, no specific signaling is required between the BS and the covered devices. Nor is it needed to perform any coordination among devices within a cluster or across different clusters.

More specifically, we propose an RL-based policy gradient (PG)-driven access control and resource allocation algorithm, that allows the BS to compute and periodically broadcast an access probability to each cluster and a hash seed [8]. In this way, access congestion is significantly reduced, as well as inter- and intra-cluster interference. In addition, the PG algorithm can be configured to improve throughput fairness among clusters located at different distances from the BS. As described later, the PG algorithm also computes *hash seeds* to dynamically support the SCF access mode [3]. The hash seeds are computed to minimize the probability that the same uplink radio resource is simultaneously selected by multiple devices from the same cluster, significantly providing an additional reduction of both inter- and intra-cluster interference. Clearly, SDP detection failures have been substantially reduced through both access control and hash-based radio resource allocation, leading to significant improvement on throughput, access delay, and energy consumption.

In brief, the novelty and main contributions of this paper are summarized as follows:

- A novel data-driven framework for clustered NOMA-enabled IoT that supports two uplink SDP access modes (SCF and CB) and an RL-enabled uplink access control and data transmission mechanism executed at the BS has been introduced.
- To support the SCF access mode, we devise a hash function-based slot selection algorithm. Contending devices compute a hash function based on two parameters, namely, the latest hash seed broadcasted by the BS and the device identity (ID). The result identifies the time slot

in a frame that the device shall use to transmit its SDP. No specific handshake between devices and the BS on channel condition or traffic state is needed.

- Relying on the *partial system state* observed by the BS, a PG-driven algorithm has been proposed to dynamically adjust access probabilities and hash seed. The PG algorithm intends to maximize system throughput or cluster throughput fairness, relying only on its observation on the number of successfully detected SDP transmissions.
- The framework has been implemented in a simulation platform. Through extensive simulations, we validate the applicability and scalability of the developed framework with different network configurations and traffic load conditions, demonstrating that quasi-optimal system performance can be achieved.
- Our performance assessment is pursued from the traffic perspective. We aim at evaluating the impact that different network operation objectives have on four performance metrics: throughput, throughput fairness, access delay, and energy consumption. We also explore and shed light on the effectiveness and scalability of the proposed PG-enabled access control mechanism.

In a nutshell, the uniqueness of this paper is represented by a combination of clustered NOMA-facilitated SDP transmission, the support of two access modes without bearing additional signaling overhead, and RL-enabled access control through online learning. The novelty of our proposal is represented by the coordinated operation of access control and radio resource allocation in a real-time manner supporting both CB and SCF access modes. The designed and implemented PG algorithm demonstrates that the RL-agent is able to efficiently and simultaneously learn, for each cluster, both an access control policy and a seed selection policy to minimize both intra- and inter-cluster interference. To the best of our knowledge, the random access and data transmission framework developed in this paper, that is empowered by RL-enabled access control with multi-dimensional decision-making, is the first effort that tackles resource allocation for uplink IoT SDP transmission with a scalable RL-enabled solution that achieves quasi-optimal performance.

The remainder of this paper is organized as follows. After presenting the envisaged network scenario and the physical layer transmission principle in Section II, the RL-enabled framework for uplink IoT data transmission consisting of two access modes is introduced in Section III. Then the core component of the framework, a PG-driven access control mechanism is proposed in Section IV with its implementation overview outlined in Section V. Section VI is dedicated to assess the performance of the developed framework with multiple network configurations and under various traffic load conditions. Furthermore, we discuss a few aspects that are related to the feasibility and operability of our framework in SectionVII. Finally, the paper is concluded in Section VIII.

II. Network Scenario and Transmission Principle

In this section, we briefly present the network scenario and the data transmission principle adopted in this study.

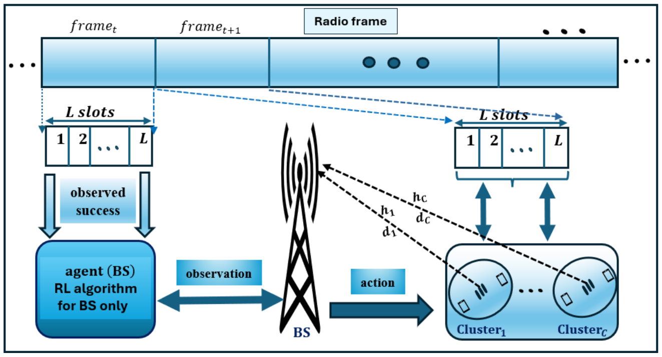


Fig. 1: Overview of the RL4SCF framework: Network scenario, frame structure, data transmission, and RL-enabled access control.

A. Network Scenario

Consider a NOMA-enabled IoT network composed of a single cell with a beamforming-assisted BS, connecting numerous devices which are uniformly distributed across the cell. Devices in the studied network are battery-powered and equipped with a single antenna. An identical transmit power level applies to all SDP transmissions and no power control is introduced [30]. For presentation simplicity, we show that a radio resource is perceived as structured in a framed slotted manner to facilitate random access and SDP transmission.

In addition, the BS is assisted by a MIMO antenna connecting numerous devices across the cell. Through beamforming and beam steering, a group of devices with an (almost) identical angle toward the BS can be covered by the same beam, leading to a less number of devices covered by one beam [4]. To cover a whole cell across multiple directions, distinct beams can be assigned. However, how to perform beamforming is beyond the scope of this paper.

Furthermore, we consider that devices are static, i.e., they do not move, and assume that no device hardware or software failure occurs during the operation of the framework. Devices are distributed into multiple clusters, where each cluster confines a number of devices that are located in the vicinity of each other. As depicted in Fig. 1, *Cluster* i (C_i), $i = 1, \ldots, C$, is composed of N_i devices and its center is located d_i meters apart from the BS. The locations of these devices may vary within the radius from the cluster center. A cluster with a lower cluster index is located closer to the BS, e.g., $d_i < d_j$ if $i < j$. A summary of the main notations used in this paper can be found in the Appendix.

B. Frame Structure and Physical Layer Principles

1) Frame structure: To facilitate flexible and efficient radio resource allocation, multiple time slots are grouped into one frame. All time slots have the same duration and a device can select at most one of the time slots in a frame to transmit its SDP. Two phases, namely, SDP transmission and acknowledgment (ACK) after a short interval upon a successful reception, occur inside one time slot. SIC is performed at the end of the data transmission phase and the ACK message is sent at the end of the time slot.

As concurrent transmissions from various devices located in the same or different clusters are allowed, one or multiple

TABLE II: Successful data detection probabilities [6]

State (n_1, n_2)	Cluster C_1				Cluster C_2			
	S_0	S_1	S_2	S_3	S_0	S_1	S_2	S_3
(0,0)	1	0	0	0	1	0	0	0
(0,1)	1	0	0	0	0.567	0.433	0	0
(0,2)	1	0	0	0	0.477	0.490	0.033	0
(0,3)	1	0	0	0	0.521	0.424	0.055	0
(1,0)	0.163	0.837	0	0	1	0	0	0
(1,1)	0.401	0.599	0	0	0.804	0.196	0	0
(1,2)	0.564	0.436	0	0	0.766	0.227	0.007	0
(1,3)	0.675	0.325	0	0	0.788	0.200	0.012	0
(2,0)	0.606	0.274	0.120	0	1	0	0	0
(2,1)	0.677	0.249	0.074	0	0.941	0.059	0	0
(2,2)	0.734	0.220	0.046	0	0.929	0.070	0.001	0
(2,3)	0.780	0.191	0.029	0	0.935	0.062	0.003	0
(3,0)	0.741	0.167	0.079	0.013	1	0	0	0
(3,1)	0.779	0.164	0.049	0.008	0.984	0.016	0	0
(3,2)	0.810	0.155	0.031	0.004	0.980	0.020	0	0
(3,3)	0.836	0.143	0.019	0.002	0.981	0.019	0	0

SDP transmissions from different devices may occur within one time slot.

2) Signal reception and channel gain: The total received signal at the BS in the k-th time slot, y_k, which is accumulated based on the individual signals from those devices that transmit concurrently within the same time slot, can be expressed as,

$$y_k = \sum_{i=1}^{C} \sum_{j=1}^{N_i} \mathcal{I}\left(i, j, k\right) \mathbf{H}_j^i x_j^i + \eta\,, \tag{1}$$

where $\mathcal{I}\left(i, j, k\right)$ is an indicator function that is 1 when the j-th device in cluster C_i transmits in the k-th time slot, and 0 otherwise. In (1), $\mathbf{H}_j^i$, x_j^i, and η represent the complex channel gain vector between device D_j^i (the j-th device in the i-th cluster) and the BS, the transmitted signal by that device, and the additive noise which follows a complex Gaussian distribution with zero mean and variance ρ^2, i.e., $\eta \sim \mathcal{CN}\left(0, \rho^2\right)$, respectively.

The channel gain between devices and the BS, which is located g meters above the ground, $\mathbf{H}_j^i \sim \mathcal{CN}\left(\mathbf{0}_M, \mathbf{I}_M\right)$, follows a Rayleigh fading model with a zero mean complex Gaussian distribution, where $M \geq 1$ is the number of antennas mounted at the BS. Moreover, the signal transmitted by device D_j^i is expressed as $x_j^i = \sqrt{P} s_j^i$, where P is the transmit power and s_j^i is the transmitted signal with unit variance. The path loss is determined by $128 + 37.6 \log_{10} d$, with d being the distance (in kilometers) between a device and the BS in a Euclidean 3D space.

3) Interference and data detection: The success or failure of an SDP transmission depends on the SINR level of the SDP signal relative to the other interfering signals in the same time slot and the noise level. To potentially detect one SDP, the signals from all other concurrent transmissions within the same slot are treated as interference to the signal that has the strongest signal strength. Applying the SIC principle for power-domain NOMA, one or more SDP transmissions received by the BS in the same time slot may be detected successfully when their SINR is greater than a given threshold, $SINR = 10$ dB, which is a realistic threshold obtained from the real-life experiments performed in [31]. Upon a successful detection, the decoded packet is subtracted from the received signals and the detection procedure proceeds successively with one less signal until all signals are processed.

Considering a cell with two clusters, we have determined the probability distributions for successful detection when multiple packets from two clusters are transmitted in the same time slot. Table II illustrates these distributions when $N_i = \{0, 1, 2, 3\}$, $i = 1, 2$, respectively. These results are reproduced from our earlier work [6], which applied the same distance and cluster radius configuration. Note that the table content for row (n_1, n_2) and column S_u is the probability that u successful packet detections occur from the corresponding cluster, when the number of packets transmitted in the same time slot are n_1 and n_2 from C_1 and C_2, respectively. Take state $(n_1, n_2) = (2, 2)$ as an example. When concurrent transmissions from four devices, two from each cluster, occur, there is a probability of 22% of detecting one packet from C_1 and of 7% of detecting one packet from C_2. Furthermore, the probabilities of two packets from the same cluster being detected successfully are 4.6% and 0.1% for $C1$ and C_2 devices respectively. Without NOMA, these probabilities would be a zero. Based on this observed benefit, we apply NOMA also in this study and reuse the probability distributions of successful detection shown in Table II.

III. RL4SCF: An RL-enabled framework

In this section, we present the developed framework as a whole and the constituent elements including access control, random access modes, hash function, and SDP transmission schemes. The RL algorithm that serves as the core for the operation of the framework will be presented later in Section IV. Four performance metrics are defined at the end of this section.

A. Framework Overview

An overview of the RL4SCF framework including the constituent elements and the operation of the framework is illustrated in Fig. 1. From a system composition perspective, the framework contains: 1) a BS that performs not only conventional 5G functions but also the new mechanism and algorithm proposed in this study; and 2) numerous IoT devices confined in two (or multiple) clusters spread randomly across the cell coverage. From a network functionality perspective, the framework 1) enhances the 5G functions by enabling RL at the BS for the purpose of achieving best possible network performance (system throughput and cluster throughput fairness); and 2) enables devices with CB or SCF access modes, supported through access control parameters periodically broadcasted by the BS.

1) RL4SCF–BS operation: The BS functions as an *RL-agent*. Based on its observation on the number of successful SDP transmissions, the BS performs RL-based optimization, takes actions on the access probability for each cluster and the hash seed that supports the SCF access mode, and broadcasts the decisions periodically to all clusters.

2) RL4SCF–device operation: Devices have a micro-controller unit (MCU) with simple computation capability. Each device operates independently from the operation of the other devices in the same cluster. *No RL capability* is required

TABLE III: Framework overview: Components, principles, & features

Component I:	**RL-enabled access control and resource allocation**
Principle and features	Executed by the BS periodically and applies to access control and slot selection for devices in both clusters that have different access probabilities

Component II:	**Random access and SDP transmission schemes**		
Principle and features	Scheme A	CB mode for C_1 devices (slots selected with equal probability)	CB mode for C_2 devices (slots selected with equal probability)
Principle and features	Scheme B	SCF mode for C_1 devices (hash-based slot selection)	CB mode for C_2 devices (slots selected with equal probability)

for the operation of devices. Nor is it needed to perform specific signaling prior to an SDP transmission by a device.

In brief, the RL4SCF framework includes two major components: 1) One access control and resource allocation *mechanism* that is enabled by an RL *algorithm*, executed at the BS; and 2) Two SDP transmission *schemes* enabled by two random access *modes*, executed by devices through the instructions by the BS. In Table III, we summarize how these components are integrated into the framework and the key features. The main elements of the framework are presented in the following subsections.

B. Access Control

Let W be the number of devices that transmit in a frame which contains L time slots. Clearly, the number of successful transmissions within the frame could be higher than L, thanks to the enhanced data detection capability provided by NOMA. Nevertheless, for a given frame length L, there exists an optimum number of devices $W^* \geq L$ that can transmit in that frame. When $W > W^*$, the likelihood of occurring data detection failure *increases*, then the number of successful transmissions *decreases*. On the other hand, radio resources are underutilized when $W < W^*$. As a measure, imposing access control leads to optimal network performance.

As a vital component of the RL4SCF framework, we propose a probabilistic access control mechanism that determines W^* *per frame and per cluster*. The access control mechanism is facilitated by an RL algorithm that runs at the BS. As the RL-agent, the BS dynamically adapts its actions based on its observation on system state and periodically broadcasts an access probability a_i[2] to C_i. The aim of the RL algorithm is to make W as close to W^* as possible, while maximizing a given performance objective as defined later in Section IV.

Devices with a non-empty queue are referred to as *active devices*. An active device in C_i generates a random variate α, uniformly distributed in $[0, 1]$, and compares it with the latest received access probability a_i from the BS. If $\alpha \leq a_i$, it will transmit in the current frame and we refer to it as an *active device* that *transmits* (ADT). Otherwise, we refer to it as an *active device* forced to *defer* (ADD) its transmission.

Within each frame, one active device may transmit *a maximum of* one packet. Active devices follow the same procedure in every frame in a memory-less manner, regardless of whether the next packet in the buffer is a newly arrived or a backlogged one. On the other hand, an ADD will attempt to transmit in the next frame with the corresponding access probability received from the BS. If a device transmits a packet but does not receive an ACK message, it will continue the same procedure until an ACK confirming the successful detection by the BS.

C. Random Access Modes

Once an active device has got access permission, it must select a single time slot in the frame to proceed with data transmission. As explained below, two access modes are defined in our framework and they constitute the basis for the SDP transmission schemes presented in the next subsection.

1) CB mode: ADT devices perform simple random selection so that a single time slot is selected *with equal probability* from the set of L slots in a frame. This mode offers basic access. However, a *slot selection collision* occurs if the same time slot in a frame is selected by multiple devices.

Depending on whether the devices that select the same time slot are from different clusters or the same cluster, a slot selection collision could lead to more serious inter- or intra-cluster interference. From our previous studies on a similar network scenario [6] [29], with one representative result shown in Table II, we conclude that intra-cluster interference appears as the most detrimental factor for NOMA-based data detection. The table also shows that avoiding slot selection collision from C_1 devices may improve C_2 SDP detections as well.

2) SCF mode: In order to diminish data detection failures, the SCF access mode is proposed. This mode is enabled by the execution of a hash function at the SCF devices. The hash function aims to distribute the time slots selected by different devices using the SCF mode as widely as possible across the frame. As such, the number of slot selection collisions caused from the SCF devices is dramatically reduced. Clearly, the SCF mode provides an access service superior to the one provided by the CB mode, leading to improved throughput, shorter delay, and reduced energy consumption.

The SCF mode is supported by an RL algorithm that runs at the BS and that periodically broadcasts a hash seed per cluster, e.g., seed b_i to C_i. An ADT determines the transmission time slot by executing a hash function that requires a hash seed b_i as input, as described next.

D. Hash-based Slot Selection

This element aims at avoiding or minimizing slot selection collision and reducing intra- and inter-cluster interference. Clearly, under heavy traffic conditions, this function cannot completely eliminate collisions, neither slot selection collisions nor data detection failures. Then, running hash-based function slot selection achieves the best performance when deployed together with an effective access control mechanism.

For each cluster, the RL-agent at the BS keeps a set of candidate seeds and finds which are the best seeds in the set, given the current state of the network. To do this, the RL algorithm learns the expected performance obtained by each seed at each system state. Note that the BS does not know the IDs of the ADTs. At a given frame, the best seed for each cluster will be the one that leads to the greatest number of successful transmissions.

The hash function f_{hash} is executed by each ADT. As the input, it takes the latest hash seed broadcasted by the BS plus the unique ID of the ADT. As the output, it returns the slot number for SDP transmission in the corresponding frame:

$$slot = f_{hash}(b_i, \mathrm{ID})\,, \tag{2}$$

[2] Note that a_i represents the fraction of active devices in C_i that will transmit in the current frame.

where b_i is the seed broadcasted to cluster C_i. As such, the slot chosen by each ADT is jointly determined by its ID and the hash seed. Then, each seed b_i represents a mapping of the ADTs in cluster C_i to the available slots.

For a given set of ADTs in C_i, the BS intends to find and broadcast a seed b_i that results in more favorable mapping of devices to slots, i.e., with fewer slot selection collisions. Note that this set of ADTs is unknown to the BS and must be estimated by the RL algorithm. The only required property of the hash function is that it must uniformly map the inputs over the output range $\{1 \dots L\}$, in order to minimize slot selection collision. This can be achieved by means of simple functions that are well suited for execution at IoT devices with low computation power through the built-in MCU at each device.

E. SDP Transmission Schemes

To facilitate random access and SDP data transmission in the envisaged network scenario, two schemes have been defined in our framework, as presented below. While Scheme A spreads the transmissions of SDPs across slots in a frame with equal probability, Scheme B focuses on diminishing slot selection collisions among C_1 devices.

1) Scheme A: Devices in both clusters, C_1 and C_2, deploy the CB mode. However, different access probabilities might be assigned to each of them by the PG-driven access control algorithm. As C_2 devices are located farther away from the BS, a higher access probability is *typically* assigned to C_2 devices to improve cluster throughput fairness among C_1 and C_2 devices.

2) Scheme B: While devices from C_2 still follow the CB mode, C_1 devices deploy the SCF mode. With the SCF mode, C_1 devices perceive much lower intra-cluster interference, leading to a higher successful SDP detection rate for C_1 transmissions. Devices from C_2 may also perceive a reduction in inter-cluster interference.

F. Performance Metrics

The following four performance metrics are defined.

- *Cluster throughput (γ_i for cluster C_i)* and *system throughput (γ_s).* They are defined as the average number of packets successfully transmitted per frame by a cluster and by the entire network including all clusters, respectively.
- *Access delay (D).* It is defined as the average number of frames it takes for a device to transmit a packet successfully. This metric encompasses not only the frames during which an ADD defers its transmission but also those frames where an SDP was transmitted but the BS failed to detect it.
- *Device energy consumption (E).* It is defined as the average energy consumed by a device per successfully transmitted SDP. It aggregates the energy consumed by an active device: i) to transmit an SDP; ii) during retransmission(s); iii) for an ACK message reception upon a successful transmission; and iv) while it defers its transmission.

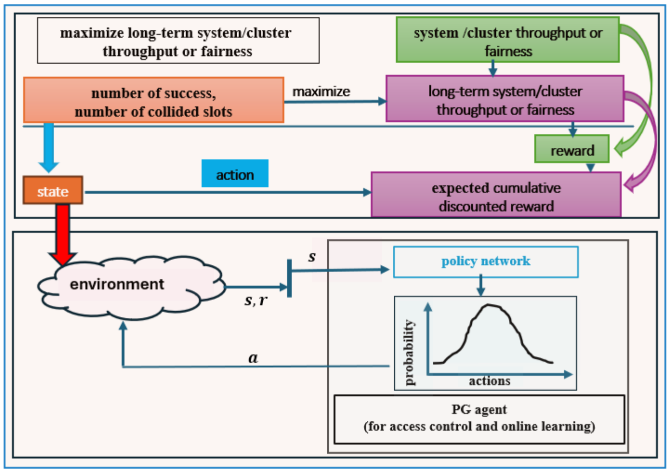

maximize long-term system/cluster throughput or fairness
system /cluster throughput or fairness
number of success, number of collided slots
maximize
long-term system/cluster throughput or fairness
reward
action
state
expected cumulative discounted reward
environment
s
s, r
a
policy network
probability
actions
PG agent (for access control and online learning)

Fig. 2: PG-driven online learning: Access control and seed generation.

- *Throughput fairness index $\widehat{J}(\gamma_1, \dots, \gamma_C)$.* It is the Jain's fairness index computed as $\widehat{J}(\gamma_1, \dots, \gamma_C) = (\sum_{i=1}^{C} \gamma_i)^2 / (C \times \sum_{i=1}^{C} \gamma_i^2)$, i.e., computed using the *average* throughput, instead of the instantaneous throughput as defined in (13). $\widehat{J}(\gamma_1, \dots, \gamma_C) \in [1/C, 1]$. Then, the closer the $\widehat{J}(\gamma_1, \dots, \gamma_C)$ to 1, the fairer the throughput distribution among clusters.

IV. Policy Gradient Supporting the Framework

PG algorithms learn parametrized policies that can select actions without consulting an action-value function [8]. The policies learned through PG are parametrized functions defining the probability of each action. In the RL4SCF framework, the PG algorithm learns two action policies for each cluster C_i: one for access probability a_i, and the other for seed b_i. Both policies are learned simultaneously by the same RL-agent.

A. States and Actions

We define the system state s as the total number of successful transmissions observed by the BS in the preceding frame, including the successful transmissions from all clusters. Let s_i denote the number of successful transmissions in cluster C_i in the previous frame. Then $s = \sum_{i=1}^{C} s_i$. The state space is discrete, $s \in \{0, \dots, \sum_{i=1}^{C} N_i\}$.

The action is a combination of a vector of access probabilities with an element a_i for every cluster i, and a vector of seeds, with an element b_i for every cluster operating in the SCF mode. The access probabilities are continuous within the interval of $[0.1, 1]$, while the seeds are taken from a discrete set of candidate seeds.

B. Policy Learning

Let $\pi_i(\cdot|s, \vec{\theta}_i)$, with parameters vector $\vec{\theta}_i$, be the access probability policy for cluster C_i, with $\pi_i(a_i|s, \vec{\theta}_i)$ being the probability of access probability a_i when the system state is s. Analogously, let $\tau_i(\cdot|s, \vec{\phi}_i)$ be the seed policy for cluster C_i, with parameters $\vec{\phi}_i$. These policies are learned through updating, at every step, the parameters vectors:

$$\vec{\theta}_i \longleftarrow \vec{\theta}_i + \alpha_\theta \, \delta \, \nabla \log \pi_i(a_i|s, \vec{\theta}_i) \, , \quad (3)$$

$$\vec{\phi}_i \longleftarrow \vec{\phi}_i + \alpha_\phi \, \delta \, \nabla \, \log \tau_i(b_i|s, \vec{\phi}_i) \, , \quad (4)$$

where α_θ and α_ϕ are learning steps. The error δ is computed from a learned state-value function $V(s,\vec{\omega})$,

$$\delta = r + \epsilon\, V(s_{\text{next}},\vec{\omega}) - V(s,\vec{\omega})\ , \tag{5}$$

where s is the current state, s_{next} the next state, r the reward and ϵ a discount rate. The state-value function is linearly approximated by $V(s,\vec{\omega}) = \omega^{(s)}$, where $\omega^{(s)}$ is the s-th element of the n-dimension vector $\vec{\omega}$, and is updated by stochastic gradient descent optimization. As all the elements of the gradient are 0 except the s-th element, which is 1, at state s only the s-th element of $\vec{\omega}$ is updated:

$$\omega^{(s)} \longleftarrow \omega^{(s)} + \alpha_\omega\, \delta\ , \tag{6}$$

where α_ω is a learning step. Policies $\pi_i(\cdot|s,\vec{\theta}_i)$ are continuous probability distributions within the range of $[0.1, 1]$. For them, we use log-normal distributions $\log a_i' \backsim \mathcal{N}(\mu_i,\sigma^2)$ transformed as

$$a_i = \frac{0.1 + a_i'}{1 + a_i'}\ . \tag{7}$$

A single value of the variance σ^2 is fixed to tune the dispersion of the distributions, while each mean μ_i is linearly approximated by $\mu_i(s,\vec{\theta}_i) = \theta_i^{(s)}$, where $\theta_i^{(s)}$ is the s-th element of the n-dimension vector $\vec{\theta}_i$. From (3), it follows that, for state s and action a_i, only the s-th element of $\vec{\theta}_i$ is updated:

$$\theta_i^{(s)} \longleftarrow \theta_i^{(s)} + \alpha_\theta\, \delta\, \frac{\log \frac{a_i - 0.1}{1 - a_i} - \theta_i^{(s)}}{\sigma^2}\ . \tag{8}$$

Policies $\tau_i(\cdot|s,\vec{\phi}_i)$ are discrete probability distributions with a value at state s for each candidate seed. We apply soft-max distributions,

$$\tau_i(b^{(j)}|s,\vec{\phi}_i) = \frac{e^{h_i(s,b^{(j)},\vec{\phi}_i)}}{\sum_b e^{h_i(s,b,\vec{\phi}_i)}}, \tag{9}$$

where $h_i(s,b^{(j)},\vec{\phi}_i) = \phi_i^{(s,j)}$ are parameterized with $n \times q$-dimension vectors $\vec{\phi}_i$, with q equal to the number of candidate seeds. For state s and seed b_i, every element (s,j) of $\vec{\phi}_i$, for $j = 1 \ldots q$, is updated as

$$\phi_i^{(s,j)} \longleftarrow \begin{cases} \phi_i^{(s,j)} + \alpha_\phi\, \delta - \tau_i(b^{(j)}|s,\vec{\phi}_i) & \text{if } b^{(j)} = b_i \\ \phi_i^{(s,j)} - \tau_i(b^{(j)}|s,\vec{\phi}_i) & \text{if } b^{(j)} \neq b_i\ . \end{cases} \tag{10}$$

C. Rewards

Two reward functions are defined. While the goal of the first reward function is to maximize the number of packets that can be successfully transmitted per frame, the second one intends to achieve certain degree of fairness among clusters.

Reward function $r^{(1)}$ is the total number of successful transmissions in the previous frame:

$$r^{(1)}(s_1,\ldots,s_C) = \sum_{i=1}^{C} s_i = s\ . \tag{11}$$

As an alternative to promote fairness in terms of throughput among clusters, we define reward function $r^{(2)}$ as

$$r^{(2)}(s_1,\ldots,s_C) = r^{(1)}(s_1,...,s_C)\ J(s_1,\ldots,s_C)\ , \tag{12}$$

```
Algorithm 1: Implementation of the PG Algorithm
Set learning steps $\alpha_\theta, \alpha_\phi, \alpha_\omega$
for each cluster $i$ do
    Initialize access probability policy $\pi_i$: $\vec{\theta}_i \leftarrow \vec{0}$
    Initialize seed policy $\tau_i$: $\vec{\phi}_i \leftarrow \vec{0}$
Initialize state-value function $V$: $\vec{\omega} \leftarrow \vec{0}$
Initialize state: $s \leftarrow 0$
for each step do
    for each cluster $i$ do
        Choose access probability $a_i \backsim \pi_i(\cdot|s,\vec{\theta}_i)$
        Choose seed $b_i \backsim \tau_i(\cdot|s,\vec{\phi}_i)$
        Broadcast $a_i$ and $b_i$
    Transmit and observe $successes$
    Set next state: $s_{\text{next}}$= $successes$
    Compute reward: $r = reward(successes)$
    Compute error: $\delta = r + \epsilon\, V(s_{\text{next}},\vec{\omega}) - V(s,\vec{\omega})$
    for each cluster $i$ do
        Update $\pi_i$: $\vec{\theta}_i \leftarrow \vec{\theta}_i + \alpha_\theta\, \delta\, \nabla \log \pi_i(a_i|s,\vec{\theta}_i)$
        Update $\tau_i$: $\vec{\phi}_i \leftarrow \vec{\phi}_i + \alpha_\phi\, \delta\, \nabla \log \tau_i(b_i|s,\vec{\phi}_i)$
    Update $V$: $\vec{\omega} \leftarrow \vec{\omega} + \alpha_\omega\, \delta\, V(s,\vec{\omega})$
    $s \leftarrow s_{\text{next}}$
```

TABLE IV: PG hyper-parameters

Parameter	Symbol	*Value*
Learning step for state-value function	α_ω	0.001
Learning step for a_i policy	α_θ	0.001
Learning step for b_i policy	α_ϕ	0.01
Number of states	n	number of total devices +1
Number of candidate seeds	q	10
a_i policy standard deviation	σ	0.1
Discount rate	ϵ	0.5

where $J(s_1,\ldots,s_C)$ is the instantaneous Jain's fairness index obtained solely based on the observed number of successes *in the preceding frame*,

$$J(s_1,\ldots,s_C) = \frac{(\sum_{i=1}^{C} s_i)^2}{C \sum_{i=1}^{C} s_i^2}\ . \tag{13}$$

V. Implementation Overview

In this section, we first summarize the implementation of the RL-enabled access control mechanism at the BS and slot selection through hashing by devices and then explain how energy consumption for a device is calculated.

A. Implementation of the PG Algorithm

Alg. 1 illustrates how PG-based access control is implemented. This algorithm is continuously running at the BS and taking actions in real-time, in a frame-by-frame manner or at a configurable update interval that may cover multiple frames. In Table IV, we present a list of hyper-parameters and their values that are adopted in our implementation.

B. Implementation of the Hash Function

The hash function (2) adopted by the devices has been implemented using the random number generator $\mathrm{rand}()$, which is a function that generates *pseudo-random integers* with minimal computational cost and this function is available

TABLE V: Parameters for energy consumption calculation [32]

Parameter	*Value*	*Unit*	*Parameter*	*Value*	*Unit*
Slot duration	20	ms	Transmit power	200	mW
Packet size	128	bytes	Reception current	35	mA
ACK size	16	bytes	Idle current	2.7	μA
Data rate	60	kbps	Voltage	3.7	volt

in any programming language. Each device at cluster C_i calculates the slot as

$$\mathrm{srand}(b_i + \mathrm{ID})\,, \tag{14}$$

$$slot = \mathrm{rand}(L), \tag{15}$$

where, as previously defined, b_i is the hash seed broadcasted by the BS to cluster C_i, ID is a unique identifier of the device, and L is the number of time slots in one frame.

To compute the time slot assignment for a device, its initial state is set to be $(b_i + \mathrm{ID})$ and the slot assigned to the device is obtained by $rand()$ mod L. This time slot assignment procedure does not lead to random mapping but achieves ultimately a hash table that maps devices to time slots within the interval of $[1, L]$, with the lowest number of collisions for the current set of transmitting devices.

C. Device Energy Consumption Calculation

We consider that a device may transmit several times across multiple frames before its transmission is acknowledged by the BS as successful through an ACK message. For a successful transmission, both SDP transmission and ACK reception occur inside one slot [4].

Denote respectively by P_{tx} and P_{rx} the transmission and the reception power for a device, T_{tx} the transmission time for one SDP, T_{rx} the reception time for ACK, T_{slot} the slot duration, N_{tot} the total number of attempts for one successful transmission among which $N_{tot} - 1$ failed and one succeeded, and N_{idle} the number of frames that an ADD defers its transmission. Then the total energy consumed by a device per successfully transmitted SDP, E_{tot}, is calculated as follows.

$$E_{tot} = P_{tx}T_{tx}N_{tot} + P_{rx}T_{rx} + P_{idle}T_{slot}LN_{idle}. \tag{16}$$

In the above expression, which does not rely on any specific type of IoT devices, we assume that the energy consumed during the interval between a successful data packet transmission and its ACK message reception inside the same time slot is negligible. Furthermore, the power consumed by a device in the idle state, P_{idle}, is typically three orders of magnitude lower than the power consumed while transmitting and receiving.

As a realistic example, we provide in Table V a list of parameters and values that are adopted in our energy consumption calculations [32]. In this example, the slot duration is configured as 20 ms as it requires (128 + 16) bytes×8/60 kbps = 19.2 ms to transmit an SDP and receive its corresponding ACK message upon a successful transmission. Note however that these parameters are configurable and the operation of our schemes is irrelevant to the slot or frame duration. The numerical results reported in Section VI are the mean values of per device energy consumption averaged over all successfully transmitted SDPs.

TABLE VI: Physical layer and network configuration

Parameter	*Value*	*Parameter*	*Value*
Number of clusters (C)	2	Standard deviation for shadow fading	8 dB
C1 to BS distance (d_1)	450 m	Receiver sensitivity	-104 dBm
C2 to BS distance (d_2)	900 m	No. of devices/cluster (N)	{8; 16; 32}
Cluster radius	25 m	No. of slots/frame (L)	{4; 8; 16}
SIC SINR threshold (β)	10 dB	Arrival probability (λ)	[0.1,...,1]
Antenna height (g)	30 m	Noise power spectral density (η^2)	-174 dBm/Hz

VI. Numerical Results and Discussions

In this section, we first explain the different network configurations that have been designed to assess the benefits that the RL4SCF framework can bring on network performance. Then, we present and discuss the performance evaluation results that are obtained through extensive simulations.

A. Network Configuration

The RL4SCF framework illustrated in Fig. 1 has been implemented based on a custom-built simulator we have developed using Java. The implemented network supports a single-cell network with a number of static devices uniformly distributed across two distinct clusters.

To evaluate the performance of the framework, we assume that packet arrivals to devices follow a Bernoulli distribution with an arrival probability *per frame* $\lambda \in [0.1, 1]$. Note that any other arrival distribution or arrival pattern may also be applied for performance evaluation.

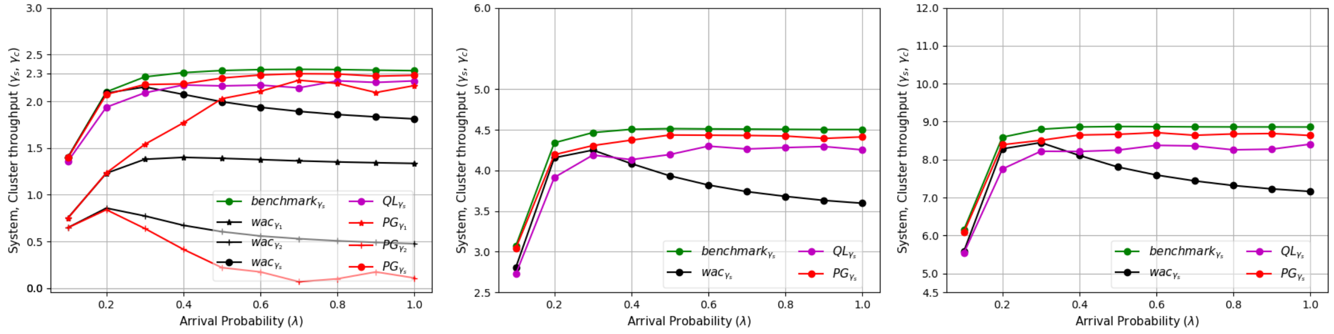
We define different network configurations, where the number of devices per cluster, N_1 or N_2, increases as the number of time slots per frame L becomes larger. For the numerical results reported in this paper, the following configuration parameters have been adopted, namely, $L = \{4, 8, 16\}$ and $N_1 = N_2 = \{8, 16, 32\}$ respectively. For simplicity, we refer to a specific network configuration by a tuple $\{L; N_1 + N_2\}$. For a complete definition of the physical layer and network configuration parameters adopted along the performance evaluation process, please refer to Table VI.

B. System and Cluster Throughput in Scheme A

In Scheme A, with both clusters operated in the CB mode, we evaluate: 1) the benefit of introducing access control; 2) how close our access control mechanism can approach an upper-bound throughput benchmark; 3) scalability of the framework; and 4) the impact that the reward function has on the pursued performance objectives.

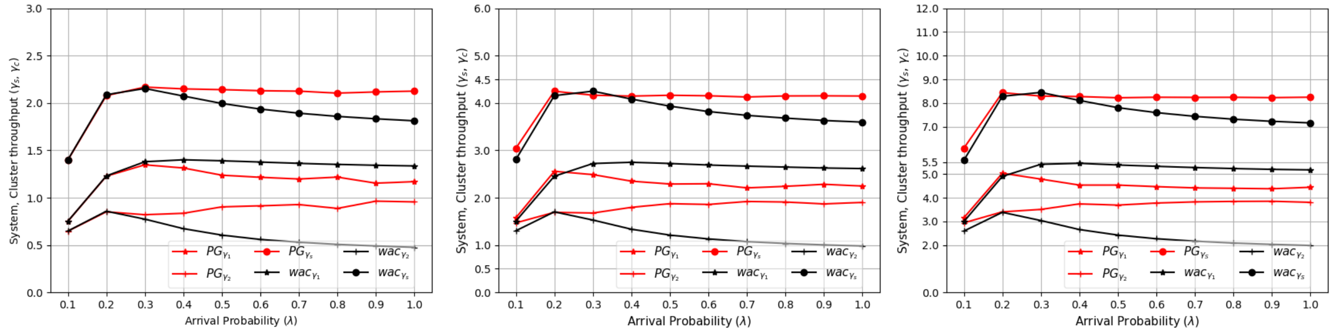
1) System throughput with and without access control: For comparison purposes, we illustrate in Fig. 3 the throughput achieved both by Scheme A and another reference scheme, referred to as *without access control* (wac), in which the access control mechanism in RL4SCF is *disabled*. For wac, neither access control nor CB or/and SCF modes are introduced.

Fig. 3 depicts the variation of the obtained system throughput (γ_s) and cluster throughput (γ_i) with the traffic load (λ). For the purpose of comparison, a curve (in violet) representing the system throughput under the same network configuration obtained from our earlier work [29] which performed a QL-based access scheme is also kept in this figure.



(a) $\{4; 8+8\}$. (b) $\{8; 16+16\}$. (c) $\{16; 32+32\}$.

Fig. 3: Throughput in Scheme A when reward function $r^{(1)}$ is adopted.



(a) $\{4; 8+8\}$. (b) $\{8; 16+16\}$. (c) $\{16; 32+32\}$.

Fig. 4: Throughput in Scheme A when reward function $r^{(2)}$ is adopted.

Before the saturation point, the achieved throughput increases with the traffic load for all the evaluated schemes. Without access control, a significant decline in system and cluster throughput is observed when $\lambda > 0.3$. Clearly, the benefit of applying access control becomes evident when the traffic load grows to a certain level ($\lambda > 0.3$), as observed in Fig. 3 for all the three network configurations.

2) System throughput versus upper-bound benchmark: The throughput upper-bound benchmark (the green curve in Fig. 3) is obtained through exhaustive search of the access probabilities that lead to the maximum system throughput for each load level.

Despite the fact that the access probability policies determined by the PG and QL algorithms are computed based on the partially observed system state information, both of them lead to superb performance. In terms of system throughput, the ones achieved by the PG algorithm proposed in this work and the QL algorithm that was obtained from [29] are quite close to the system throughput benchmark which is the performance upper-bound.

3) Throughput fairness: In Fig. 4, we focus on the throughput achieved by each cluster for the same network configurations studied above. In this figure, we also include the system throughput as a reference and refer to the cluster throughput achieved by RL4SCF as PG. As it is evident, the difference between the cluster throughput of the two clusters becomes much smaller when reward function $r^{(2)}$ is adopted, compared to the difference between the ones shown in Fig. 3a), where function $r^{(1)}$ was used. Correspondingly, the fairness indexes $\widehat{J}(\gamma_1, \gamma_2)$ for $\lambda = 1.0$ achieved in Fig. 4 are 0.998, 0.996, and 0.996 for the three studied network configurations, $\{4; 8+8\}$, $\{8; 16+16\}$, and $\{16; 32+32\}$, respectively. These values indicate that with $r^{(2)}$ near-perfect throughput fairness between the two clusters has been achieved, demonstrating the effectiveness of the proposed PG algorithm.

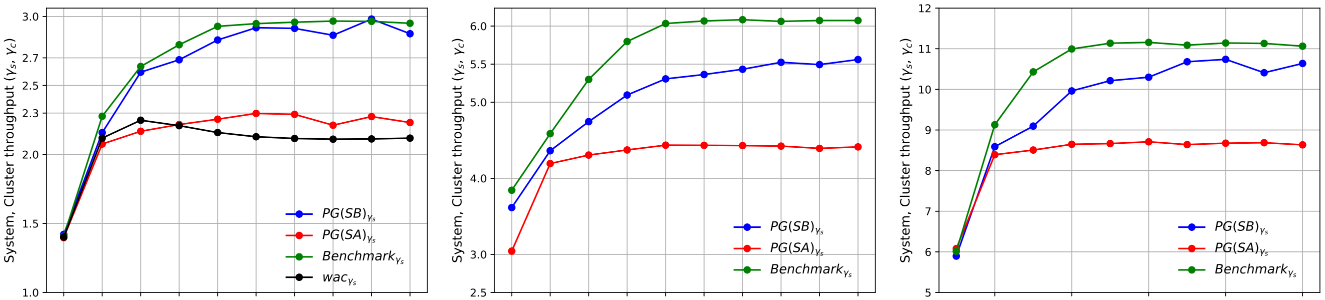
In contrast, the fairness indexes[3] shown in Fig. 3 that are obtained for the same network configurations when $\lambda = 1.0$ are 0.545, 0.559, and 0.560, respectively. Clearly, when the goal of the PG algorithm is to maximize system throughput, the throughput fairness is considerably penalized.

However, when the PG algorithm intends to maximize cluster throughput fairness using $r^{(2)}$, the system throughput in the three studied network configurations when $\lambda = 1.0$ are, respectively, 8.26%, 6.01%, and 6.78% lower than the ones obtained based on reward function $r^{(1)}$ under the same traffic load. In other words, cluster throughput fairness is achieved at the cost of merely a minor system throughput decrement.

C. System and Cluster Throughput in Scheme B

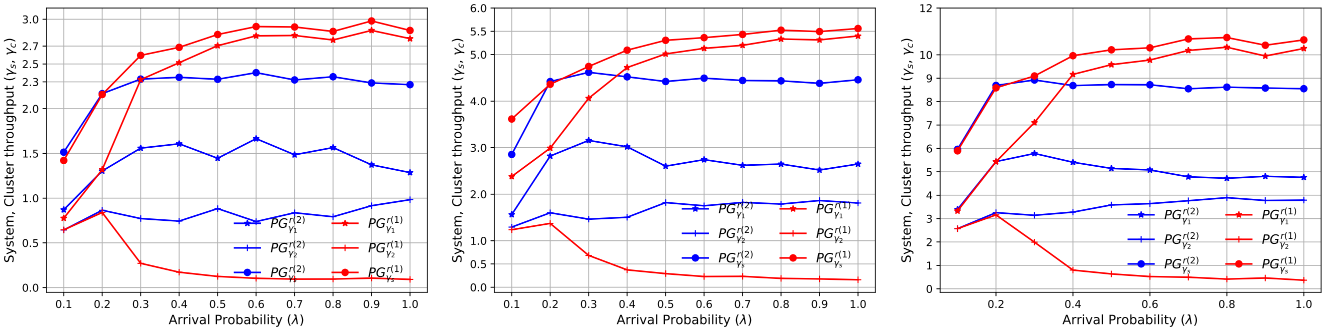
Having investigated the performance of the PG-driven access control mechanism in terms of throughput and fairness in Scheme A, we focus in this subsection on evaluating the impact that hashing has on system performance in Scheme B, with cluster C_2 in the CB mode and cluster C_1 in the SCF mode. As above, the metrics used to evaluate this impact are throughput and throughput fairness. The results are displayed in Fig. 5 and Fig. 6, respectively.

[3] For illustration clarity, we do not depict the cluster throughput in Fig.3b) and Fig.3c). Nor in Fig.5b) and Fig.5c).



(a) $\{4; 8 + 8\}$. (b) $\{8; 16 + 16\}$. (c) $\{16; 32 + 32\}$.

Fig. 5: Throughput in Scheme B when using reward function $r^{(1)}$.



(a) $\{4; 8 + 8\}$. (b) $\{8; 16 + 16\}$. (c) $\{16; 32 + 32\}$.

Fig. 6: Throughput in Scheme B when using reward functions $r^{(1)}$ and $r^{(2)}$.

1) The benefit of hashing on system throughput: Let us first assess the benefit brought by the hash function when employing the radio resource allocation algorithm supported by the execution of a hash function at the devices, with the system throughput for all the three network configurations illustrated in Fig. 5. In this figure, by PG(SA) (in red) and PG(SB) (in blue) we refer to the system throughput achieved by Scheme A and B, respectively, when reward function $r^{(1)}$ is deployed. As in Scheme A, the green curve in this figure represents a system throughput benchmark in Scheme B when $r^{(1)}$ is adopted. Here, the throughput *upper-bound* benchmark has been obtained by exhaustive search of the access probabilities and the hash seeds, assuming that the BS has complete knowledge of the system state and the IDs of all ADTs in the cell. This knowledge allows the BS to configure the access probability and seed values in a way that maximize system throughput. However, it is noteworthy to clarify that this benchmark represents *an ideal* upper-bound on system performance rather than an optimal solution, as it cannot be achieved in a real-life deployment scenario.

In Fig. 5a), we still keep the curve labeled as *wac* to show that the proposed PG-driven access control mechanism brings a huge benefit. Clearly, the system throughput in Scheme B is substantially higher then the one in Scheme A. In particular, it is 28.70%, 26.65%, and 24.30% higher for the three network configurations when $\lambda = 1.0$, respectively, contributing significantly to improved total system throughput.

2) Throughput fairness: In Fig. 6, we compare the cluster throughput fairness achieved in Scheme B when deploying reward functions $r^{(1)}$ and $r^{(2)}$. We denote by $PG^{r(n)}_{\gamma_m}$, $n = 1, 2$, $m = s, 1, 2$, the corresponding throughput achieved for the whole system, by cluster C_1 or C_2, respectively.

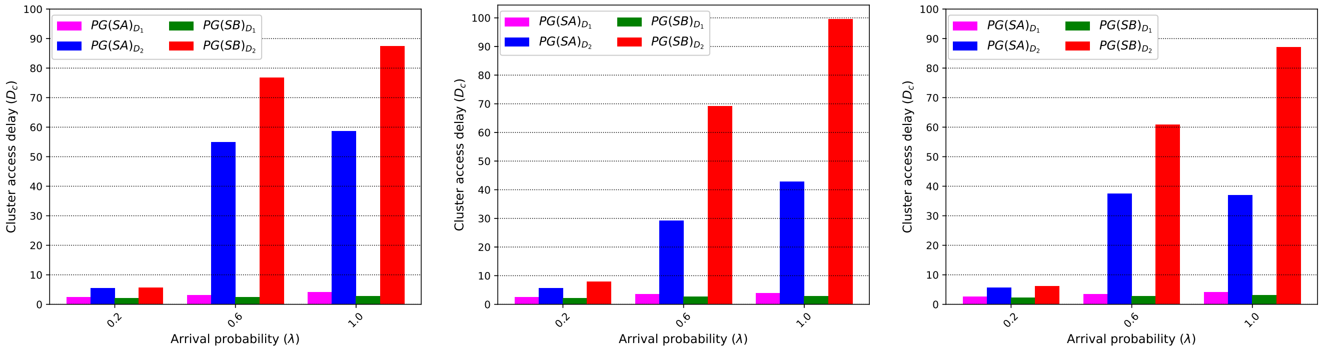
Clearly, with $r^{(1)}$, higher system throughput and C_1 cluster throughput are achieved. On the other hand, when $r^{(2)}$ is adopted, we obtain much better throughput fairness. More specifically, the achieved fairness index values with $r^{(2)}$ are 0.98, 0.97, and 0.99 for the three network configurations when $\lambda = 1.0$, respectively.

However, the improved throughput fairness is obtained at the cost of lower system throughput. When deploying reward functions $r^{(2)}$, the observed system throughput reductions when $\lambda = 1.0$ are 20.90%, 19.78%, and 19.64% , respectively, in comparison with the ones obtained when deploying $r^{(1)}$. Note, however, that the system throughput with $r^{(2)}$ is higher in Scheme B than in Scheme A. In general, which reward function to adopt depends on the service requirements.

D. Access Delay

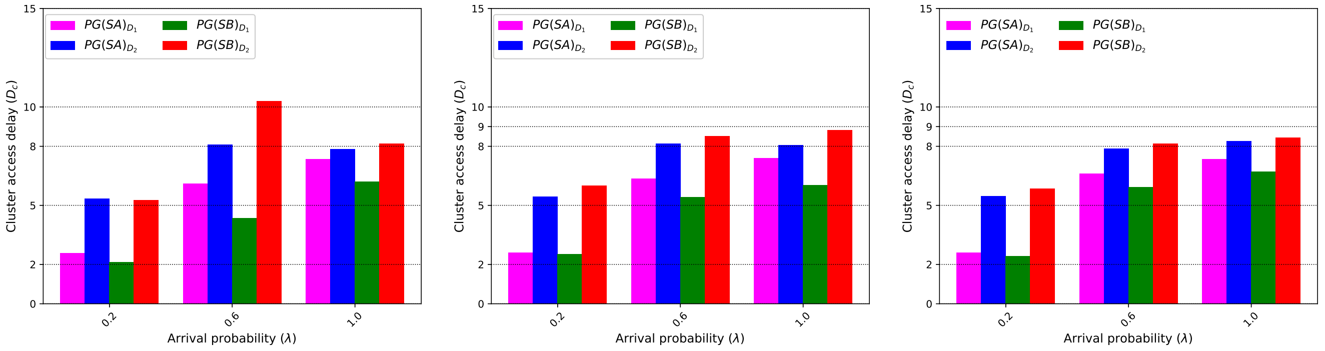
1) Access delay when reward function $r^{(1)}$ is adopted: As defined in (11), the main goal of reward function $r^{(1)}$ is to maximize system throughput, regardless of the achieved throughput fairness index. In pursuit of this goal, the PG algorithm assigns a higher access probability to C_1 devices than to C_2 devices, as the first ones are closer to the BS than the second ones. Then, C_1 devices achieve lower and more stable delays under all traffic load conditions and across all network configurations, as shown in the pink and green bars in all three sub-figures in Fig. 7. On the other hand, devices in C_2, with a lower access probability, perceive longer delays per successfully SDP transmission.

When comparing the access delay achieved in Schemes A and B, it is clear that C_2 devices achieve comparatively longer delays in Scheme B than in Scheme A. As explained above, the RL algorithm assigns an even higher access probability to C_1 devices in Scheme B than in Scheme A, as C_1 devices in



(a) Delay $\{4; 8+8\}$ (b) Delay $\{8; 16+16\}$ (c) Delay $\{16; 32+32\}$

Fig. 7: Access delay when using reward function $r^{(1)}$ under light, medium, and heavy traffic load.



(a) Delay $\{4; 8+8\}$ (b) Delay $\{8; 16+16\}$ (c) Delay $\{16; 32+32\}$

Fig. 8: Access delay when using reward function $r^{(2)}$ under light, medium, and heavy traffic load.

Scheme B adopt the SCF access mode, resulting in substantially improved successful detection rate. This leads to a small delay reduction for packets from C_1 devices in Scheme B than in Scheme A. However, packets from C_2 devices in Scheme B experience a substantially longer delay than in Scheme A.

2) Access delay when reward function $r^{(2)}$ is adopted: Different from $r^{(1)}$, reward function $r^{(2)}$ defined in (12) aims to achieve throughput fairness across clusters. Accordingly, balanced access delays among devices from different clusters may be achieved.

When comparing the results in Fig. 7 and Fig. 8, we reveal that much shorter access delay differences between clusters have been achieved by using reward function $r^{(2)}$ than when using $r^{(1)}$, in both evaluation schemes and for all the investigated network configurations and traffic load conditions. Take network configuration $\{8; 16+16\}$ and medium traffic load, $\lambda = 0.6$, as an example. In Scheme A, C_1 and C_2 devices achieve an access delay (marked in pink and blue in Fig. 7b) and Fig. 8b), respectively) of 3.51 and 29.22 frames with $r^{(1)}$, versus 6.36 and 8.15 frames with $r^{(2)}$. However, this delay fairness improvement is achieved at the expense of C_1 devices experiencing longer delay. In other words, 3.58 times shorter delay has been experienced by C_2 devices at the cost of 1.81 times longer delay experienced by C_1 devices. With the same network configuration $\{8; 16+16\}$ and traffic load $\lambda = 0.6$ but using Scheme B, where C_1 and C_2 devices deploy the SCF and CB modes, respectively, devices in C_1 enjoy 2.00 times shorter access delay than C_2 devices at the cost of 8.12 times longer access delay for C_2 devices (marked in green and red in Fig. 7b) and Fig. 8b), respectively).

E. Device Energy Consumption

In Table VII, we present the average energy consumed by a device per successfully SDP transmission, where E_1 and E_2 represent the per device energy consumption averaged across all the devices in C_1 and C_2 and frames throughout the simulation, respectively. In general, a C_1 device consumes less energy than a C_2 device does for all the investigated network configurations and traffic load conditions. Clearly, C_1 devices, being closer to the BS, require less number of attempts to complete a successful SDP transmission than C_2 devices do.

For a fixed traffic load, the energy consumed by both C_1 and C_2 devices increases moderately with network size. Although the ratio of the number of radio resources (time slots) in a frame per device is maintained identically for all the studied network configurations, more active devices per frame in a larger network induce higher interference.

1) Energy consumption, Scheme A versus Scheme B: Comparing the energy consumption in Schemes A and B for a given network size and load, we reveal that, in Scheme B, C_1 device consumption is lower, while C_2 device consumption is higher than in Scheme A. In Scheme B, C_1 devices employ the SCF mode, enjoy less intra-cluster interference and, then, less transmission attempts per successful SDP transmission, resulting in less energy consumption. On the other hand, the fact that C_1 devices are closer to the BS creates higher interference on C_2 devices, than vice versa. This effect is more acute when the BS runs $r^{(1)}$ as, in Scheme B, the PG algorithm perceives a higher successful detection rate for C_1 devices and increases its access probability with respect to Scheme A.

TABLE VII: Per device energy consumption (in mJ) in Scheme A

λ	Reward function $r^{(1)}$						Reward function $r^{(2)}$					
	$\{4; 8+8\}$		$\{8; 16+16\}$		$\{16; 32+32\}$		$\{4; 8+8\}$		$\{8; 16+16\}$		$\{16; 32+32\}$	
	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2
0.2	8.75	18.03	9.30	18.71	10.21	19.44	8.80	18.12	9.34	18.74	10.26	19.51
0.6	10.87	23.91	12.20	23.37	13.84	26.01	11.29	24.30	12.53	25.77	12.98	26.03
1.0	12.53	25.00	13.66	27.34	15.41	28.03	11.89	25.90	12.28	26.31	13.19	27.19

TABLE VIII: Per device energy consumption (in mJ) in Scheme B

λ	Reward function $r^{(1)}$						Reward function $r^{(2)}$					
	$\{4; 8+8\}$		$\{8; 16+16\}$		$\{16; 32+32\}$		$\{4; 8+8\}$		$\{8; 16+16\}$		$\{16; 32+32\}$	
	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2	E_1	E_2
0.2	7.28	17.29	8.15	19.55	9.57	19.96	7.54	17.76	9.13	20.58	11.84	26.70
0.6	8.90	32.70	10.77	29.01	12.75	27.89	9.30	25.47	10.30	27.35	11.78	27.19
1.0	9.30	33.10	11.43	39.25	13.97	36.52	9.24	24.80	10.94	27.66	12.09	27.86

Take network configuration $\{8; 16+16\}$, running $r^{(1)}$ and medium traffic load, $\lambda = 0.6$, as an example. Let $N^X_{tot}(r^{(i)}, C_i)$, $X = A, B$, $i = 1, 2$, be the average number of required transmission attempts per successful SDP transmission for a C_i device in Scheme X when the BS runs $r^{(i)}$. We obtain that $N^A_{tot}(r^{(1)}, C_1) = 3.17$ and $N^B_{tot}(r^{(1)}, C_1) = 2.62$. Then, the energy consumed by C_1 devices in Scheme B is 11.7% lower than in Scheme A. We show also that $N^A_{tot}(r^{(1)}, C_2) = 6.79$ and $N^B_{tot}(r^{(1)}, C_2) = 8.46$. Then, the energy consumed by C_2 devices in Scheme B is 24.6% higher than in Scheme A.

As mentioned previously, the energy consumed by C_2 devices is higher than the one consumed by C_1 devices. For $\{8; 16+16\}$ and $\lambda = 0.6$, we find that, in Scheme A, $N^A_{tot}(r^{(1)}, C_1) = 3.17$ and $N^A_{tot}(r^{(1)}, C_2) = 6.79$. Then, the energy consumed by C_2 devices is 91.6% higher than the one consumed by C_1 devices. We also illustrate that $N^B_{tot}(r^{(1)}, C_1) = 2.62$ and $N^B_{tot}(r^{(1)}, C_2) = 8.46$, and the energy consumed by C_2 devices is 169.4% higher than the one consumed by C_1 devices in Scheme B.

2) Energy consumption, $r^{(1)}$ versus $r^{(2)}$: In Scheme A, the energy consumed by C_1 and C_2 devices does not change significantly no matter the BS runs $r^{(1)}$ or $r^{(2)}$, except for large network sizes, where the energy consumption slightly decreases when the BS runs $r^{(2)}$ instead of $r^{(1)}$. In Scheme B, the energy consumed by C_1 devices follows a similar trend.

The energy consumed by C_2 devices in Scheme A increases when the BS runs $r^{(2)}$ instead of $r^{(1)}$. However, in Scheme B, it decreases when the BS runs $r^{(2)}$ instead of $r^{(1)}$. When the BS runs $r^{(2)}$, the objective of the PG algorithm is to maximize cluster throughput fairness, and this is achieved by reducing the access probability to C_1 devices. In Scheme A, the higher power level of C_1 with respect C_2 SDP signals received by the BS, together with the random time slot selection of active C_1 and C_2 devices, have a higher impact on the inter-cluster interference perceived by C_2 devices than the fact that C_1 devices are assigned a lower access probability. However, in Scheme B, C_1 devices deploy the SCF access mode that helps to evenly spread SDP transmissions over different time slots. As a consequence, C_2 devices perceive less inter-cluster interference and their successful detection rate increases when the BS runs $r^{(2)}$ instead of $r^{(1)}$.

Take network configuration $\{8; 16+16\}$ and medium traffic load, $\lambda = 0.6$, as an example. We find that, in Scheme A, $N^A_{tot}(r^{(1)}, C_1) = 3.34$ and $N^A_{tot}(r^{(2)}, C_1) = 3.43$. Then, the energy consumption of C_1 devices when the BS runs $r^{(2)}$ is 2.7% lower than when running $r^{(1)}$. We reveal further that, in Scheme B, $N^B_{tot}(r^{(1)}, C_1) = 2.62$ and $N^B_{tot}(r^{(2)}, C_1) = 2.51$. Then, the energy consumption of C_1 devices when the BS runs $r^{(2)}$ is 4.4% lower than when running $r^{(1)}$.

Moreover, we illustrate that, in Scheme A, $N^A_{tot}(r^{(1)}, C_2) = 6.79$ and $N^A_{tot}(r^{(2)}, C_2) = 7.36$. Then, the energy consumption of C_2 devices when the BS runs $r^{(2)}$ is 10.3% higher than when running $r^{(1)}$. For comparison, we obtain in Scheme B $N^B_{tot}(r^{(1)}, C_2) = 8.46$ and $N^B_{tot}(r^{(2)}, C_2) = 7.56$. Then, the energy consumption of C_2 devices when the BS runs $r^{(2)}$ is 5.7% lower than when running $r^{(1)}$.

In summary, the most favorable configuration for reducing energy consumption is Scheme B with the BS running $r^{(2)}$. Here, the energy consumed by C_1 devices is the lowest one. While the energy consumed by C_2 devices is approximately the same as in Scheme A, either running $r^{(1)}$ or $r^{(2)}$, it is lower than the one consumed in Scheme B with the BS running $r^{(1)}$.

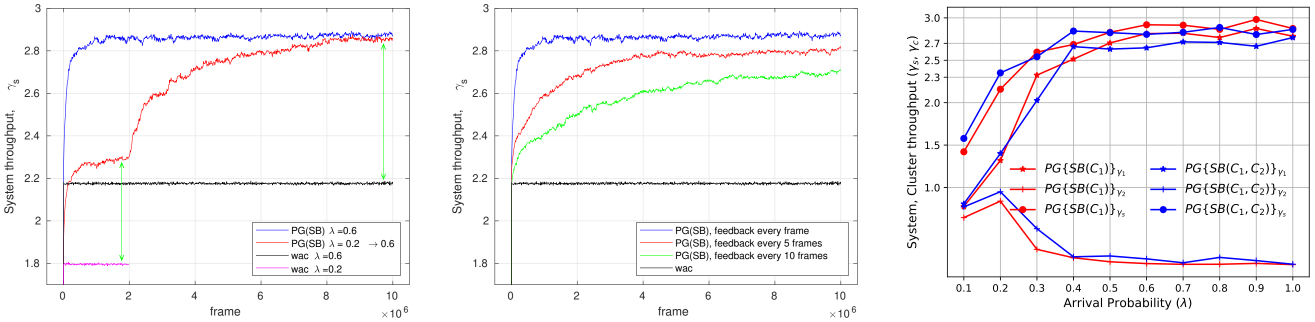
VII. Feasibility and Operability

In this section, we further explore six other aspects that may have an effect on or are related to the feasibility and operability of the developed RL4SCF framework.

A. Optimality

By comparing the system throughput achieved in Scheme A with the system throughput benchmark in Fig. 3, we conclude that quasi-optimal performance has been achieved by the proposed PG algorithm. When observing the corresponding results for Scheme B in Fig. 5, we notice that the achieved system throughput is below the benchmark for some network configurations. This result is in accordance with the operation principle of Scheme B, as multi-dimensional policies have to be learned online simultaneously when running both SCF and CB, making the optimization problem more challenging with a larger network size.

Considering that the PG-agent in our framework is purely model-free and actions are taken solely based on the agent's



(a) System throughput over time from cold start and abrupt transition: Reward function $r^{(1)}$

(b) System throughput over time with 3 different updating intervals: Reward function $r^{(1)}$

(c) System and cluster throughput: Hashing applies to both clusters: Reward function $r^{(2)}$

Fig. 9: Scheme B for network configuration $\{4; 8+8\}$: (a) System throughput from a cold start; (b) System throughput with different updating intervals; (c) Throughput: Hashing for both.

partial knowledge on system state, we ascertain that feasibility, operability, and excellent performance have been achieved by the proposed PG-driven access control and resource allocation mechanism for SDP transmissions.

B. Convergence

To provide an insight into the convergence time of the RL-enabled control mechanism, we reveal in Fig. 9a) the evolution of the average system throughput over time, obtained based on Scheme B, from a cold start and also from an abrupt transition of the traffic load. The average throughput is measured in a time window of 10^4 frames, and the plot illustrates the average values based on 10 runs. The blue line is the throughput from a cold start, with the policy parameters $\vec{\theta}_i$ and $\vec{\phi}_i$ set to 0, and with the arrival probability to all devices set, for consistency and uniformity, to $\lambda = 0.6$. The red line is the system throughput from a cold start with $\lambda = 0.2$ until frame $2 \cdot 10^6$, the moment at which $\lambda = 0.6$.

From a cold start, more than 90% of the gain is achieved around the $2 \cdot 10^5$ frame, and after that the gain converges slowly to the maximum. This can be seen in the blue line, for the high load case, and in the red line up to frame $2 \cdot 10^6$, for low load. The green arrows show the maximum throughput gains relative to transmission without access control. As can be observed, the system takes longer time to converge from a previously learned behavior than from a start with the policy parameters set to zero. However, it is worth noting that this experiment examines an extreme case, in which the load changes instantaneously on all devices from a very low value to a very high value. In a more realistic scenario, the system load would likely change more smoothly.

C. Periodicity of BS Broadcasts

In the numerical results presented above in Section VI, the BS broadcasts an access probability and a hash seed per cluster once every $\mathcal{K} = 1$ frame. However, it is worth clarifying that the BS broadcast periodicity in the RL4SCF framework can be set to any other value by appropriately configuring $\mathcal{K}$.

As an example, we investigate in Fig. 9b) how a longer updating and broadcast interval would affect performance and convergence time, by illustrating the evolution of the average system throughput over time in Scheme B, for updating intervals configured as $\mathcal{K} \in \{1, 5, 10\}$ frames respectively. The results reveal that the algorithm converges faster with a shorter updating and broadcast interval, i.e., a smaller $\mathcal{K}$.

D. Complexity and Scalability

In a general sense, the complexity of the PG algorithm (Alg. 1) is affected by two factors related to network size, as explained below:

- Number of time slots. Let $\mathcal{S}$ be the state space and $|\mathcal{S}|$ its cardinality. Then, $|\mathcal{S}| = fL+1$, with L being the number of time slots allocated for SDP transmissions and f being the maximum number of SDP successful detections per time slot. Note that f will take a small value as in the studied network configurations. For each cluster C_i, the access probability policy $\pi_i(\cdot|s, \vec{\theta}_i)$ has to maintain and update a parameter vector $\vec{\theta}_i$ with $|\mathcal{S}|$ elements. Each seed policy $\tau_i(\cdot|s, \vec{\phi}_i)$ has to maintain and update a parameter vector $\vec{\phi}_i$ with $|\mathcal{S}| \times q$ elements, with q being the number of candidate hash seeds. Also, the parameter vector $\vec{\omega}$ in the state-value function $V(s, \vec{\omega})$ has $|\mathcal{S}|$ elements.
- Number of clusters C. For each cluster, two policies are required: π_i for the access probability and τ_i for the seed for cluster C_i.

Additionally, each of these policies is composed of a set of *independent* policies, one for each cluster. Thus, the domain of each of the functions to be learned (the policies) is discrete and one-dimensional, with size n (number of states), which in turn depends solely on the number of active devices. Therefore, the complexity of Alg. 1, in terms of memory size and processing time, *grows only linearly* both with the number of time slots and with the number of clusters, demonstrating the scalability of the developed RL-enabled mechanism for access control and data transmission of uplink IoT traffic.

In contrast, *conventional RL methods* require to learn and use a state-action-value function, and the domain dimension of this function grows with the number of clusters. Furthermore, the action space must be discretized. Therefore, the number of elements in the domain of the state-value function would be $n \times P^C \times q^C$, where q is the number of candidate seeds and P is the number of access probabilities resulted in from the discretization of the interval $[0\ 1]$. Furthermore, the selection of actions requires a search, which is generally not efficient, in the state-action-value function. Consequently, these methods exhibit higher complexity in terms of both memory size and processing time, and this complexity potentially grows exponentially with the size of the problem, limiting their scalability.

E. Applying Hashing to C2 Devices

We are also interested in investigating the performance of the proposed framework when the SCF mode is applied to C_2 devices as well. This is because we expect that a hashing-based slot selection by C_2 devices will lead to significantly lower intra-cluster interference and slightly less inter-cluster interference to C_1 transmissions.

When reward function $r^{(1)}$ is adopted, the results are almost identical to those shown in Fig. 5 that are obtained when only C_1 devices deploy the SCF mode. The reason for this behavior is straightforward. The BS receives weaker signals from C_2 devices than from C_1 devices. Accordingly, the successful SIC detection rate of SDPs from C_2 devices is largely dependent on the absence of any concurrent transmission from C_1 devices. Then, even when the activities of C_2 devices produce lower intra- and inter-cluster interference, its impact is faded away by the signal strength of C_1 devices. Clearly, when $r^{(1)}$ is adopted, C_1 devices will still be assigned a much higher access probability than C_2 devices.

In this line, we expect that when reward function $r^{(2)}$ is adopted for the purpose of improving throughput fairness, the impact of configuring C_2 devices with the SCF mode will be more noticeable. Taking network size $\{4; 8 + 8\}$ as an example, we illustrate in Fig. 9c) the obtained cluster and system throughput. The curves in blue (labeled as $PG\{SB(C_1, C_2)\}\gamma$, etc.) represent the throughput when both C_1 and C_2 devices run the SCF mode, whereas the curves in red (labeled as $PG\{SB(C_1)\}\gamma$, etc.) represent the throughput when only C_1 devices run SCF. With $r^{(2)}$, C_1 devices are assigned lower access probabilities than with $r^{(1)}$. Then, with lower concurrent transmission from any C_1 device, slot allocation through hashing substantially reduces intra-cluster interference and increases the successful SDP detection rate from C_2 devices, as shown in the γ_2 curves in Fig. 9c).

Although not shown explicitly herein, an additional benefit of running the SCF mode in both clusters is a reduction of energy consumption. On average, C_1 and C_2 devices observe a relative energy consumption reduction of 20.5% and 18.4%, respectively.

F. Applicability to Multiple Clusters

This study focuses on a network scenario considering that devices in a cell covered by the same beam are confined into two clusters and our access control and SDP transmission schemes deal with both intra- and inter-cluster interference. When more than two clusters exist in the same cell, inter-cluster interference would increase significantly. Specifically, the performance of devices located in the farthest cluster(s) from the BS would likely deteriorate due to higher path loss combined with increased interference. As a consequence, it would become much more difficult for the SIC procedure to effectively distinguish and detect data packets successively.

Such a problem in NOMA networks is indeed well known in the research community. To solve this problem, various *user pairing* and *clustering* strategies have been proposed (see e.g., [18] [20] [34] [35] [36] [37]). Once coupled with a proper cluster paring algorithm, the proposed framework is inherently scalable and can be extended to accommodate multiple clusters in larger network deployments. However, developing any other cluster pairing strategy or algorithm beyond the location-based clustering is outside the scope of this paper.

VIII. Conclusions and Future Work

In this paper, we have presented an RL-enabled framework for SDP transmission tailored to uplink traffic in clustered NOMA-facilitated IoT networks, supporting hybrid contention-based and semi-contention-free access modes. As the core component of the framework, the BS, as an RL-agent, performs online learning through a policy gradient algorithm. It computes access probabilities for both access modes to maximize system throughput or achieve throughput fairness among clusters. Also, it computes hash seeds to support the semi-contention-free access mode operation. When a device operates in the semi-contention-free mode, it reduces both intra- and inter-cluster interference. To operate the RL4SCF framework, no assumption on the state of the system (number of active devices) is required at the BS. Nor is it necessary for the BS and devices to perform any protocol handshake prior to a data transmission. IoT devices only need to execute a lightweight hash function and to perform simple computations based on the seed and access probability that they receive from the BS periodically.

By illustrating various numerical results in terms of four performance metrics, we showcase not only the operability and efficiency but also the scalability and feasibility of the RL-enabled solution for random access and SDP transmission in B5G IoT networks. The findings of this study include: 1) It is beneficial to integrate SCF with CB for SDP transmission and which combination to apply depends on service requirements; 2) The proposed framework exhibits robust and stable performance, being able to achieve quasi-optimal or excellent performance in the investigated network configurations; and 3) The complexity of our developed framework is low and the access modes scale well with network size. For devices, light computation capacity suffices. As our future work, we regard integrating more precise physical layer models, developing cluster pairing algorithms, and applying other RL algorithms into the framework as potential directions.

Appendix
Summary of Notations and Descriptions

References

[1] M. Vaezi, A. Azari, S. R. Khosravirad, M. Shirvanimoghaddam, M. M. Azari, D. Chasaki, and P. Popovski, "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 4, pp. 1117–1174, 2nd Quart., 2022.

[2] *Framework and overall objectives of the future development of IMT for 2030 and beyond*, Recommendation ITU-R M.2160-0, ITU-R, Nov. 2023.

[3] C.-A. Hsu, C.-H. Tsai, F. Y. Li, C.-Y. Chen, and Y.-C. Tseng, "Receiver-initiated data collection in wake-up radio enabled mIoT networks: Achieving collision-free transmissions by hashing and partitioning," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 868-883, Jun. 2021.

[4] E. Dahlman, S. Parkvall, and J. Sköld, *5G NR: The Next Generation Wireless Access Technology*, 2nd ed., London, UK: Academic Press, 2021.

[5] T. N. Weerasinghe, V. Casares-Giner, I. A. M. Balapuwaduge, and F. Y. Li, "Priority enabled grant-free access with dynamic slot allocation for heterogeneous mMTC traffic in 5G NR networks," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 3192–3206, May 2021.

Notation	**Description**
C	Total number of cluster
C_i	Cluster i
D_j^i	Device j in the cluster C_i
N_i	Total number of devices in Cluster C_i
y_n	Total received signal at the BS in the n-th time slot
$\mathcal{I}\left(i,j,n\right)$	Indicator function that is 1 when D_j^i transmits in the n-th time slot, and 0 otherwise
$\mathbf{H}_j^i$	Complex channel gain vector between D_j^i and BS
x_j^i	Transmitted signal by device D_j^i
W	Number of devices that actually transmit in a frame
L	Total number of time slots in a frame
E	Device energy consumption
a_i	Access probability for cluster C_i
b_i	Seed for cluster C_i
γ_i	Throughput for cluster C_i
γ_s	Total system throughput
$J(...)$	Jain's fairness index
s	System state
s_i	Total number of successful transmissions in C_i
s_{next}	System state in the next frame
r	Immediate reward received after transition
$\pi_i(a_i \mid s,\vec{\theta}_i)$	Probability of access probability a_i for state s
$\vec{\theta}_i$	Parameters vector of the access policy for cluster C_i
$\tau_i(b_i \mid s,\vec{\phi}_i)$	Probability of selecting seed b_i in state s for C_i
$\vec{\phi}_i$	Parameters vector of the seed policy for cluster C_i
$\phi_i^{(j)}$	j-th element of the parameter vector $\vec{\phi}_i$
α_θ	Learning rate for updating $\vec{\theta}_i$
α_ϕ	Learning rate for updating $\vec{\phi}_i$
δ	Temporal-difference error or learning signal
$V(s,\vec{\omega})$	State-value function at state s with parameters $\vec{\omega}$
$\vec{\omega}$	Parameter vector for value function, dimension n (number of states)
$\omega^{(s)}$	s-th element of vector $\vec{\omega}$; value estimate for state s
α_ω	Learning rate for updating $\vec{\omega}$
q	Number of candidate seeds for each cluster
$h_i(s,b_i,\vec{\phi}_i)$	Score function used in the softmax, often set as $\phi_i^{(s)}$
$e^{h_i(\cdot)}$	Exponential of the score for softmax normalization

[6] A. Kumar, J. Martinez-Bauset, F. Y. Li, C. Florea, and O. A. Dobre, "Understanding inter- and intra-cluster concurrent transmissions for IoT uplink traffic in MIMO-NOMA networks: A DTMC analysis," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14328–14343, Apr. 2024.

[7] *Study on enhancements for artificial intelligence (AI)/machine learning (ML) for NG-RAN*, Technical Report TR38.743 R19, v19.0.0, 3GPP, Sep. 2024.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., Denver, CO, USA: Bradford Books, 2018.

[9] *NR and NG-RAN Overall Description*, Technical Specification TS38.300 R18, v18.5.0, 3GPP, Mar. 2025.

[10] 3GPP, "Small Data Transmission (SDT)", 21 Jun. 2023, [Online]. Available: https://https://www.3gpp.org/technologies/sdt/.

[11] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1805–1838, 3rd Quart., 2020.

[12] I. N. A. Ramatryana and S. Y. Shin, "Priority access in NOMA-based slotted ALOHA for overload 6G massive IoT," *IEEE Commun. Lett.*, vol. 26, no. 12, pp.3064-3068, Dec. 2022.

[13] M. Liu, J. Zhang, K. Xiong, M. Zhang, P. Fan, and K. B. Letaief, "Effective user clustering and power control for multiantenna uplink NOMA transmission," *IEEE Wireless Trans. Commun.*, vol. 21, no. 7, pp. 8995–9009, Nov. 2022.

[14] L. Valentini, E. Bernardi, F. Saggese, M. Chiani, E. Paolini, and P. Popovski, "Contention-based mMTC/URLLC coexistence through coded random access and massive MIMO," *IEEE J. Sel. Areas Signal Process.*, vol. 18, no. 7, pp. 1265–1280, Oct. 2024.

[15] Z. Ding, R. Schober, P. Fan, and H. V. Poor, "Simple semi-grant-free transmission strategies assisted by non-orthogonal multiple access," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4464–4478, Jun. 2019.

[16] Z. Ding, R. Schober, and H. V. Poor, "A new QoS-guarantee strategy for NOMA assisted semi-grant-free transmission," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7489–7503, Nov. 2021.

[17] A. Rech, S. Tomasin, L. Vangelista, and C. Costa, "Semi-grant-free orthogonal multiple access with partial-information for short packet transmissions," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 3000–3013, Nov. 2023.

[18] A. S. Rajasekaran and H. Yanikomeroglu, "Neural network aided user clustering in mmWave-NOMA systems with user decoding capability constraints," *IEEE Access*, vol. 11, pp. 45672–45690, May 2023.

[19] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan, "Resource allocation in uplink NOMA-IoT networks: A reinforcement-learning approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5083–5098, Aug. 2021.

[20] J. Ren, Z. Wang, M. Xu, F. Fang, and Z. Ding, "An EM-based user clustering method in non-orthogonal multiple access," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8422–8434, Dec. 2019.

[21] X. Liu, H. Ding, and S. Hu, "Uplink resource allocation for NOMA-based hybrid spectrum access in 6G-enabled cognitive Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15049–15060, Oct. 2021.

[22] M. V. da Silva, S. Montejo-Sánchez, R. D. Souza, H. Alves, and T. Abrão, "D2D assisted Q-learning random access for NOMA-based MTC networks," *IEEE Access*, vol. 10, pp. 30694–30706, Mar. 2022.

[23] J. Zhang, X. Tao, H. Wu, N. Zhang, and X. Zhang, "Deep reinforcement learning for throughput improvement of the uplink grant-free NOMA system," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6369–6379, Jul. 2020.

[24] J. Liu, Z. Shi, S. Zhang, and N. Kato, "Distributed Q-learning aided uplink grant-free NOMA for massive machine-type communications," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2029–2041, Jul. 2021.

[25] Y. Liu, Y. Deng, H. Zhou, M. Elkashlan, and A. Nallanathan, "Deep reinforcement learning-based grant-free NOMA optimization for mURLLC," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1475–1490, Mar. 2023.

[26] R. Kumar, M. Gupta, S. Jha, and T. Singh, "Double deep Q-learning for dynamic resource allocation in NOMA IoT," *IEEE Access*, vol. 11, pp. 12345–12359, May 2023.

[27] F. Liu, J. Zhang, H. Zhou, and K. Wang, "Multi-agent reinforcement learning for grant-free NOMA in dense IoT networks," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4000–4013, Jun. 2022.

[28] L. Chen, Y. Wu, Z. Sun, and M. Elkashlan, "RL-based power control for grant-free uplink NOMA in massive IoT," *IEEE Commun. Lett.*, vol. 26, no. 4, pp. 876–880, Apr. 2022.

[29] A. Kumar, J. Martinez-Bauset, and F. Y. Li, "Dynamic medium access in clustered NOMA IoT networks based on reinforcement learning," in *Proc. IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2024, pp. 1–6.

[30] Z. Ding, R. Schober, and H. V. Poor, "Unveiling the importance of SIC in NOMA systems – Part 1: State of the art and recent findings," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2373–2377, Nov. 2020.

[31] Y. Qi, X. Zhang, and M. Vaezi, "Over-the-air implementation of NOMA: New experiments and future directions," *IEEE Access*, vol. 9, pp.135828–135844, Sep. 2021.

[32] *nRF9160 cellular IoT system-in-package*, nRF9160 Product Specification v2.2, Nordic Semiconductor, Jun. 2024. [Online]. Available: https://docs.nordicsemi.com/.

[33] *NR; Radio Resource Control (RRC); Protocol Specification*, Technical Specification TS38.331 R18, v18.4.0, 3GPP, Dec. 2024.

[34] Z. Ding, F. Adachi, and H. V. Poor, "The application of MIMO to non-orthogonal multiple access," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 537–552, Jan. 2016.

[35] W. A. Al-Hussaibi and F. H. Ali, "Efficient user clustering, receive antenna selection, and power allocation algorithms for massive MIMO NOMA systems," *IEEE Access*, vol. 7, pp. 31865–31882, Feb. 2019.

[36] A. S. Rajasekaran, O. Maraqa, H. U. Sokun, H. Yanikomeroglu, and S. Al-Ahmadi, "User clustering in mmWave-NOMA systems with user decoding capability constraints for B5G networks," *IEEE Access*, vol. 8, pp. 209949–209963, Nov. 2020.

[37] A. Shahini and N. Ansari, "NOMA aided narrowband IoT for machine type communications with user clustering," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7183–7191, Aug. 2019.