

## Article

# RL-SCAP SigFox: A Reinforcement Learning Based Scalable Communication Protocol for Low-Power Wide-Area IoT Networks

Raghad Albalawi <sup>1,2,\*</sup> , Fatma Bouabdallah <sup>3,\*</sup> , Linda Mohaisen <sup>1</sup>  and Shireen Saifuddin <sup>1</sup> 

<sup>1</sup> Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; ssaifuddin@kau.edu.sa (S.S.)

<sup>2</sup> College of Computer Science and Engineering, University of Hail, Hail 81481, Saudi Arabia

<sup>3</sup> School of Engineering, Computing and Mathematics, University of Plymouth, Plymouth PL4 8AA, UK

\* Correspondence: ralbalawi0022@stu.kau.edu.sa or r.albalawi@uoh.edu.sa (R.A.); fatma.bouabdallah@plymouth.ac.uk (F.B.)

**Abstract:** The Internet of Things (IoT) aims to wirelessly connect billions of physical things to the IT infrastructure. Although there are several radio access technologies available, few of them meet the needs of Internet of Things applications, such as long range, low cost, and low energy consumption. The low data rate of low-power wide-area network (LPWAN) technologies, particularly SigFox, makes them appropriate for Internet of Things applications since the longer the radio link's useable distance, the lower the data rate. Network reliability is the primary goal of SigFox technology, which aims to deliver data messages successfully through redundancy. This raises concerns about SigFox's scalability and leads to one of its flaws, namely the high collision rate. In this paper, the goal is to prevent collisions by switching to time division multiple access (TDMA) from SigFox's Aloha-based medium access protocol, utilizing only orthogonal channels, and eliminating redundancy. Consequently, during a designated time slot, each node transmits a single copy of the data message over a particular orthogonal channel. To achieve this, a multi-agent, off-policy reinforcement learning (RL) Q-Learning technique will be used on top of SigFox. In other words, the objective is to increase SigFox's scalability through the use of Reinforcement Learning based time slot allocation (RL-SCAP). The findings show that, especially in situations with high node densities or constrained communication slots, the proposed protocol performs better than the basic SCAP (Slot and Channel Allocation Protocol) by obtaining a higher Packet Delivery Ratio (PDR) in average of 60.58%, greater throughput in average of 60.90%, and a notable decrease in collisions up to 79.37%.

**Keywords:** optimization; Low-Power Wide-Area Networks (LPWAN); multiagents; transmission parameters; reinforcement learning; SigFox technology; scalability; internet of things (IoT)



Academic Editor: Spyridon Nikolaidis

Received: 21 April 2025

Revised: 4 June 2025

Accepted: 9 June 2025

Published: 17 June 2025

**Citation:** Albalawi, R.; Bouabdallah, F.; Mohaisen, L.; Saifuddin, S. RL-SCAP SigFox: A Reinforcement Learning Based Scalable Communication Protocol for Low-Power Wide-Area IoT Networks. *Technologies* **2025**, *13*, 255. <https://doi.org/10.3390/technologies13060255>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Internet of Things (IoT) has the potential to change our way of life and work by providing its fundamental services to all connected smart devices. IoT solves several wireless network problems, such as a large number of smart devices with reliable internet access, energy-constrained portable edge devices, quick data transfers, efficient bandwidth use, and improved data collection accuracy at the receiver. To accomplish these goals, the devices must be effectively connected so that they can exchange sensed data rapidly and make wise decisions to regulate physical phenomena, which will lead to the creation of a smart ecosystem. Numerous studies predict that the Internet of Things

will be crucial to a wide range of commercial and social applications in the years to come, such as smart buildings, utilities, consumer electronics, smart cities, smart transportation, smart healthcare, security, logistics, climate-smart agriculture, asset tracking, and waste management systems. Over 75 billion connected devices are expected to exist, as seen in Figure 1, based on an analysis by the Statista Research Department (SRD). Numerous wireless technologies, including long-range cellular networks and short-range Wireless Sensor Networks (WSNs), have been developed to assist the Internet of Things. Short-range wireless technology is a subset of wireless technologies used on the Internet of Things. However, there are some challenges with short-range wireless solutions for IoT networks, including weak network resilience, restricted network scalability, and increased network development costs. In contrast, the Internet of Things, which depends on cellular networks, has a complex infrastructure, a short network lifetime, and a high cost of network development. Long-range and short-range wireless communication methods for Internet of Things applications face the aforementioned difficulties, which have resulted in the development of Low Power Wide Area Networks (LPWANs). The IoT goal is met by the widely used and leading LPWAN wireless Wide Area Network (WAN) technology, which has wide coverage, low-energy consumption, effective bandwidth utilization, and low implementation costs for networks. Several wireless communication technologies are compared for their energy effectiveness and implementation costs in Figure 2. With the least expensive implementation and optimum energy efficiency, LPWAN seems to be the ideal choice. LPWAN typically offers up to forty kilometers of wide coverage in rural areas with strong Line of Sight (LoS) and ten kilometers in urban areas with weak LoS, with a 10-year minimum battery life [1,2]. One of the most important criteria for LPWAN systems is the ability to accommodate a large number of devices sending small quantities of traffic across long distances. The increasing densities of linked devices must be supported by these technologies. In our proposal, we will select SigFox, one of the best LPWAN technologies for deployment. In Europe and Asia, SigFox is extensively used [3]. The main advantage of Sigfox is its resistance to collisions and interferences, which is enabled by the use of a diversity mechanism that takes into account both time and frequency. As a result, each sensor sends a data packet at three randomly selected times via three communication channels. This approach adds redundancy at the communication level, but it has a major disadvantage because it increases channel occupancy and, implicitly, the chance of collisions. [4]. Reinforcement Learning (RL) offers a promising approach to addressing the scalability challenges of Low-Power Wide-Area Networks (LPWANs), such as SigFox and LoRa. As LPWANs accommodate an increasing number of devices, managing limited resources like bandwidth and communication slots becomes critical. RL enables systems to adapt dynamically by learning optimal strategies for resource allocation and collision avoidance through trial-and-error interactions with the network environment. By leveraging RL, LPWAN systems can efficiently assign communication slots, reduce collisions, and improve metrics like throughput and Packet Delivery Ratio (PDR), even as the number of connected devices scales up. This adaptability ensures network reliability while maintaining energy efficiency, making RL a key tool in enhancing the performance and scalability of LPWANs in IoT ecosystems. Our contribution is to propose a novel Reinforcement Learning (RL) based enhancement of the Slot and Channel Allocation Protocol (SCAP) [5], called RL-SCAP. By introducing an adaptive slot allocation technique, this protocol addresses issues with interference and resource usage by enabling nodes to dynamically learn the best communication schedules. Particularly:

1. Improved Slot Allocation: By using RL to enable autonomous slot allocation based on collision input, RL-SCAP dramatically lowers collision rates and boosts network efficiency. Each node independently observes the outcome of its previous transmis-

sions and uses this local feedback to adapt its slot selection policy. The protocol aims to achieve a mutually exclusive slot assignment among nodes, where convergence is reached when no collisions occur. This decentralized learning process is guided by lightweight feedback from the gateway, ensuring informed decision-making without requiring full centralization.

2. **Performance Optimization:** Through simulations, we thoroughly assess RL-SCAP, demonstrating how well it improves important metrics, including throughput, collision rate, and packet delivery ratio (PDR) across a variety of network settings. The adaptability of the protocol to dynamic environments ensures consistently better performance compared to static slot allocation.
3. **Scalability and Energy Efficiency:** The protocol is appropriate for Internet of Things applications involving battery-constrained devices since it is made to scale effectively in big networks while consuming the least amount of energy through highly reduced collisions and retransmissions. RL-SCAP maintains low complexity and overhead, making it suitable for LPWANs with limited resources.
4. **Hybrid Distributed Architecture:** RL-SCAP introduces a lightweight, partially decentralized architecture in which nodes make local decisions based on observations and minimal guidance from the gateway. This hybrid setup allows efficient convergence to collision-free communication while maintaining low complexity and minimal overhead, making it suitable for real-world LPWAN deployments.

These contributions show how RL-SCAP can be used to solve problems with scalability, efficiency, and reliability in dynamic Internet of Things environments.

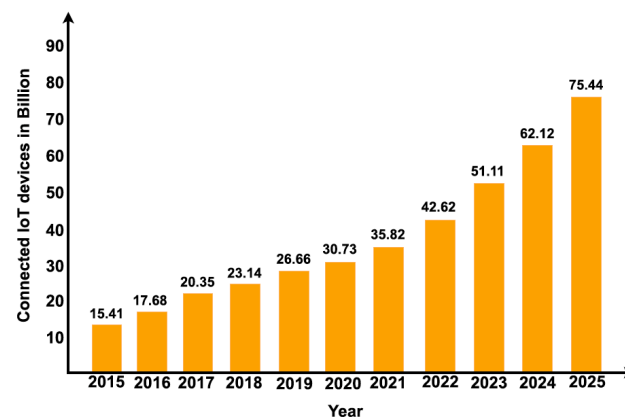


Figure 1. Growth of IoT devices during the years 2015 and 2025.

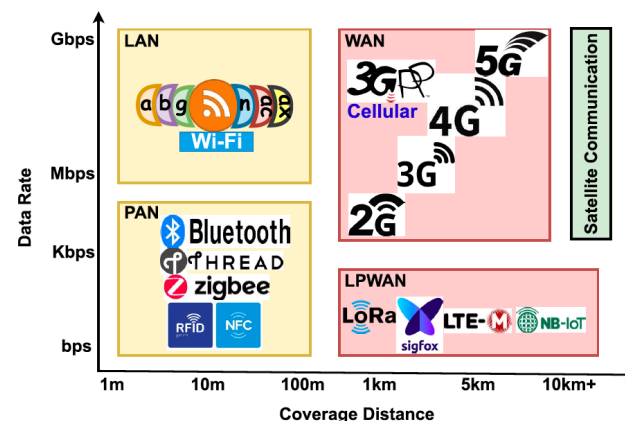


Figure 2. Comparison between different wireless technologies.

The rest of this paper is organized as follows: Section 2 examines Related Work, classifies AI and non-AI solutions, designed to address LPWAN scalability issues. Section 3 addresses the scalability problems in SCAP and the requirement for improved slot allocation mechanisms. Section 4 introduces the RL-SCAP protocol and describes its reinforcement learning architecture and slot allocation method. Section 5 includes the experimental setup, measures, and results. Finally, Section 6 concludes the paper and highlights future research directions.

## 2. Related Work

One of the most important criteria for LPWAN systems is the ability to accommodate a large number of devices sending a small amount of traffic across long distances. The increasing densities of connected devices must be supported by these technologies. The growing number of connected devices in Low-Power Wide-Area Networks (LPWANs) creates scalability issues that can result in resource congestion, decreased reliability, and degraded performance. Innovations in network architecture and design at several levels are needed to address these problems. There are mainly two approaches to address scalability challenges facing Low Power Wide Area Networks (LPWANs), according to recent research: artificial intelligence (AI) and non-AI methods.

### 2.1. Non-AI Techniques

Using dynamic resource allocation strategies is one efficient strategy. It has been demonstrated that adaptive spreading factor (SF) allocation based on signal-to-noise ratio (SNR) and lightweight scheduling techniques increases network capacity and scalability. The authors in [6,7] have emphasized that these techniques maximize channel use and guarantee equitable access for devices in dense deployments. Furthermore, load-balancing and clustering techniques can disperse traffic among gateways, reducing bottlenecks and enhancing system performance. Another important element in improving scalability is interference management. Methods, including adaptive transmission power regulation, frequency hopping, and time variety have been shown to be successful in lowering collisions and guaranteeing reliable communication. The study conducted in [1] stresses that adaptive power control, in particular, reduces interference while preserving the necessary quality of service. Another possible approach is the application of concurrent and multi-band transmission technology. According to [8], throughput and capacity can be greatly increased by permitting simultaneous operation over various frequency bands. Additionally, by decreasing the load on individual gateways, multi-hop communication protocols can promote scalability and increase coverage. Scalability is also critically dependent on energy efficiency. By using edge computing and data aggregation techniques, intermediate nodes or gateways can preprocess data, lowering the volume of packets sent and saving energy. For further improving energy efficiency while preserving communication quality, there are duty-cycling methods, which turn on devices only when required, as proposed in [2]. Network management strategies, such as centralized monitoring platforms and distributed architectures, provide scalability by enabling dynamic reconfiguration and real-time adjustments based on traffic patterns and device activity. These platforms facilitate the proactive management of resources and improve the system's adaptability to changing conditions [3]. Employing these strategies guarantees that LPWANs can accommodate the expanding needs of Internet of Things applications while also resolving their scalability problems. LPWANs may support extensive deployments without sacrificing performance by utilizing developments in network management, interference reduction, and resource optimization. On the other hand, traditional techniques like simulation, traffic analysis, and protocol design are usually used in non-AI procedures. In the study conducted in [9],

optimized traffic-aware channel assignment and contention algorithms in LoRa-based IoT networks without the usage of artificial intelligence (AI) were proposed. The simulation and modeling of LoRa and Sigfox LPWAN technologies were also investigated by the work proposed in [10], which offered insightful information about how these technologies function under different circumstances. The work proposed in [11] used traditional analytical techniques to do a scalability analysis of the Sigfox protocol in high-density, large-scale scenarios. Further illustrating the ongoing value of non-AI approaches in tackling network issues is a collision-prevention mechanism for IoT networks that was presented by the IEEE Internet of Things Journal (2019) and used planning techniques to lower network collisions. For fundamental network research and protocol development, non-AI approaches are still crucial, even though AI techniques have more promising optimization capabilities. A noteworthy study that tackles LPWAN scaling issues without using AI approaches is the one designed in [12]. The adaptive mechanism for modifying the Time-Slotted LoRa MAC protocol's frame size is proposed in this research, allowing for better scalability and more effective resource utilization. The method improves performance without the need for sophisticated machine learning techniques by concentrating on dynamic frame adaptation based on traffic circumstances. To increase the scalability in IoT networks, a significant study designed in [13] suggests a revolutionary time-slotted LoRa MAC protocol. The key innovation is an adjustable frame size technique that adjusts to network traffic conditions, ensuring better scalability and more efficient use of resources in large-scale LoRa installations. This technique enhances network performance by dynamically changing the communication parameters without the hassle of machine learning. The protocol's drawbacks include the possibility of less-than-ideal performance in extremely dynamic networks, where real-time adjustments could react more slowly to abrupt changes in traffic. Furthermore, in highly heterogeneous systems, the absence of centralized control over network resource management may result in inefficiencies. Notwithstanding these difficulties, the protocol provides a workable, non-AI way to increase the scalability and effectiveness of LoRa-based LPWANs.

## 2.2. AI Techniques

Artificial intelligence (AI) techniques like deep reinforcement learning (DRL) and reinforcement learning (RL) have been widely employed to enhance network performance. Reinforcement Learning (RL) is increasingly being implemented to solve scalability challenges in Low Power Wide Area Networks (LPWANs), particularly by optimizing resource allocation and energy efficiency. For example, RL-based methods, represented as a Markov Decision Process (MDP), can dynamically choose transmission parameters, including energy levels, coding rates, and spreading factors. These techniques allow nodes to adjust to changes in the environment, such as residual energy levels and channel statuses, which lowers long-term energy expenditures while enhancing network scalability and reliability. This method provides more energy efficiency and longer device lifespans than conventional techniques such as LoRaWAN's Adaptive Data Rate (ADR) [14,15]. Another RL-based method for reducing bottleneck energy consumption in multi-hop LPWAN configurations is centralized routing optimization. These systems use algorithms such as  $\epsilon$ -greedy to dynamically explore routing paths in order to maximize the efficiency and lifespan of the network. Large-scale LPWANs require these strategies since the network's expansion can quickly increase the number of routing options, which could otherwise result in inefficiencies as proven in [14]. In order to enhance the efficiency of massive IoT installations, the protocol presented in [16] suggested a collision avoidance strategy for distributed huge IoT systems that uses LoRa signals and is based on reinforcement learning (RL). By superposing LoRa signals, their method minimizes collisions and enables devices to learn the



best transmission parameters. Although the technique shows notable benefits in reducing collisions, the RL model's convergence time and computational complexity may limit its effectiveness, especially in dense networks with a large number of devices. Similarly, RL was used in [17] to improve the distribution of network resources in LPWANs. Their approach dynamically modifies resource allocation to enhance system performance and scalability by utilizing reinforcement learning agents. The method might, however, have trouble adjusting to extremely dynamic settings where network conditions shift quickly, which could cause training to become more complex and possibly cause delays in reaching optimal policies. RL was used by the author in [18] to enhance medium access control (MAC) in LoRa networks. By using adaptive MAC techniques, their technique successfully decreased collisions and increased network throughput. However, in situations where there is little feedback or little opportunity for node-to-node communication, the RL-based MAC protocol may have excessive latency during the learning phase and struggle to strike a balance between exploration and exploitation. This was further developed in [19] who used DRL to optimize the Spreading Factor (SF) allocation in LoRa networks, which is essential for enhancing the overall throughput and scalability of the network.

SigFox is one of the most widely deployed LPWAN technologies in Europe and Asia [3]. The primary advantage of Sigfox is its resistance to collisions and interferences, which is enabled by the use of a diversity mechanism that considers both time and frequency. Each data packet is thus sent by each sensor on three communication channels at three arbitrarily chosen times. This mechanism involves adding communication-level redundancy; however, it also comes with a significant drawback due to an increase in channel occupancy and, implicitly, the likelihood of collisions [4].

In this work, we will provide an improved version of the Slot and Channel Allocation Protocol (SCAP) [5] that uses the Q-learning technique to enhance the slot allocation procedure: the Reinforcement Learning-Based Slot and Channel Allocation Protocol (RL-SCAP).

### 3. Problem Statement

LPWAN encompasses a number of technologies, including SigFox, Long Range (LoRa), Narrowband-IoT (NB-IoT), Ingenu RPMA, Long Term Evolution for Machines (LTE-M), DASH7, Telensa and others. The aforementioned technologies have similar operational principles, but they differ in how they are implemented and behave. While unlicensed LPWAN technologies, such as LoRa, Sigfox, and RPMA, perform exceptionally well in terms of battery life, network capacity, and cost, licensed LPWAN technologies, such as NB-IoT, LTE-M, and EC-GSM-IoT, are more effective in terms of service quality (QoS), reliability, latency, and range. Therefore, the requirements of IoT applications must be taken into consideration when choosing the right LPWAN technology. The capacity to support a large number of devices sending small amounts of traffic across long distances is one of the most crucial requirements for LPWAN technology. The increasing densities and quantity of connected devices must be supported by these technologies. However, the LPWAN technologies that have been developed up to this point face difficulties with medium access mode, duty cycle, and the optimal distribution of transmission parameters that severely restrict the scalability of the network. Sigfox, one of the most famous and deployed LPWAN technologies, uses the ALOHA access mode, which is simple since it is a random-access protocol that allows any device to send a packet at any time. Despite its simplicity, when there are several devices, it has been demonstrated that ALOHA raises the collision rate. In addition to the medium access mode, duty cycling is another main scalability constraint of Sigfox technology. Although it helps regulate access to the common channel and reduce energy consumption, duty cycling may prevent the optimal configuration of the network. For instance, in Sigfox, due to the use of the free unlicensed ISM band, the duty cycle will

help share the free band among several technologies. However, it will highly constrain downlink communication that may be targeted to optimally configure the network for better performance and better scalability. Indeed, Sigfox provides the possibility to send multiple redundant copies of the same packet over a large number of narrowband channels. The non-optimal configuration of the redundancy factor and the selected channels for transmission may increase the collision rates and the packet error rates, which will end up highly reducing the scalability of Sigfox technology. For instance, the Sigfox gateway can link hundreds of thousands of end devices; however, studies [6–8,11] show that performance declines as the number of devices rises over a few hundred. Therefore, scalability is a critical issue for Sigfox networks.

#### 4. Protocol Description

Sigfox is a well-known Low-Power Wide-Area Network (LPWAN) technology that enables long-range, energy-efficient communication for Internet of Things networks. Sigfox exploits narrowband communication to provide scalability and large coverage while operating in the unlicensed sub-GHz ISM band. In the European band, SigFox uses a sub-GHz band with a width of 192 kHz, which produces 1920 partially overlapping channels with a width of 100 Hz each, of which only 360 are orthogonal. However, issues like channel allocation and collision avoidance, which are essential for guaranteeing dependable network performance, are brought on by the increasing density of connected devices.

The Slot and Channel Allocation Protocol (SCAP) is a protocol designed for wireless communication systems, particularly in Sigfox, to optimize resource utilization in dense IoT networks. SCAP tackles the challenges of limited communication resources by assigning both time slots and communication channels to devices based on their geographical locations, to ensure collision-free communication to the maximum possible extent [5].

SCAP assumes that the base station is located at the center of a circular network field. The field is partitioned into sectors, each with an angle  $\alpha$ , as seen in Figure 3, where nodes in the same sector are assigned the same channel. In fact,  $\alpha$  can be expressed easily as follows if  $M$  channels are to be used:

$$\alpha = 2\pi/M \quad (1)$$

Note that since the SCAP goal was to use every orthogonal channel, then  $M = 360$  and, hence,  $\alpha = 2\pi/360$ .

After determining the angle  $\alpha$ , the network radius, represented by  $R$ , is the maximum distance between the center of the network (gateway or base station) and its furthest node. The following formula can be used to find the area of a sector with an angle  $\alpha$ :

$$SectorArea = \frac{\alpha}{2} \times R^2 \quad (2)$$

In SCAP, the network operates within organized time frames, each subdivided into multiple time slots. These time slots are allocated to devices in the same sector to ensure collision-free communication, as no two devices transmit simultaneously on the same channel. Time slot assignment typically follows a static or predetermined schedule, simplifying network coordination but limiting adaptability to dynamic traffic conditions.

This dual-layered approach of assigning time slots and orthogonal channels prevents collisions and enhances network scalability by efficiently distributing communication resources among devices [5].

- **Channel Allocation:** A communication channel is associated to every sector. Specific channels are assigned to devices in order to prevent interference. Based on the device's

position relative to the base station, SCAP first assigns a sector identifier before allocating a channel to each device.

- **Time Slot Allocation:** Devices in the network are assigned specified time slots during which they are authorized to transmit data. Devices that are closer to the base station transmit first, preventing any collisions, because devices are given a time slot according to their distance from the base station. The following formula establishes the maximum number of slots  $m$  in a given sector.

$$m = \frac{R}{d} \quad (3)$$

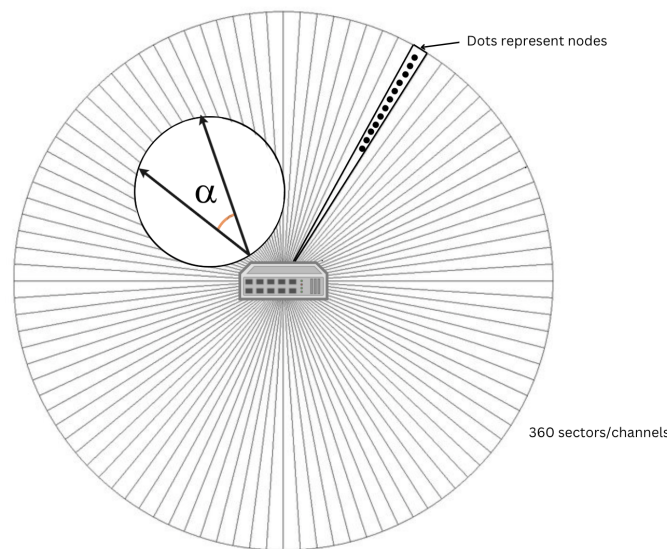
where

- $d$  is the average distance between the IoT devices

Each device is assigned a time slot based on its distance from the base station. The time slot for device  $n$ , denoted by  $T_n$ , is computed as follows:

$$T_n = \left\lceil \frac{D_n}{d} \right\rceil \quad (4)$$

where  $D_n$  is the distance between device  $n$  and the base station. This ensures that devices further away from the base station have later transmission slots, reducing the risk of collisions.



**Figure 3.** The Sector Angle of RL-SCAP.

While SCAP effectively reduces collisions and interference, its static time slot allocation mechanism is less suited to dynamic environments where the network conditions frequently change. Moreover, SCAP nodes in the same sector that are within the same distance from the gateway will end up using the same slot, and hence, collisions will be experienced. These limitations are addressed in RL-SCAP by introducing reinforcement learning, enabling dynamic and adaptive slot assignment tailored to real-time network conditions [5].

In order to overcome the drawbacks of static allocation and improve the network performance of SigFox, particularly in extremely dynamic environments with fluctuating node densities and unpredictable traffic patterns, in this work, we proposed a Reinforcement Learning-Based Slot and Channel Allocation Protocol (RL-SCAP). RL-SCAP is a machine learning extension of the classic Slot and Channel Allocation Protocol (SCAP). SCAP provides a straightforward, deterministic method for channel and time slot allocation, but it is not flexible enough to optimize communication in real-time as network conditions



change. By using Reinforcement Learning (RL) to automatically allocate communication time slots based on dynamic observations, RL-SCAP increases network throughput, packet delivery ratio (PDR), and energy efficiency while minimizing interference and collisions. The main advantage of RL-SCAP over SCAP is its ability to learn from experience and adapt to changing network conditions. Every device in the network uses Q-learning to automatically choose the optimal timeslots for its transmissions based on local observations and historical data, continuously enhancing its slot selection method. In addition to allowing for the effective use of available slots, this RL-based method helps the protocol manage the scalability issues that large-scale IoT networks confront, where static allocation would become unfeasible and inefficient. In contrast to SCAP, which uses preset time slot allocation, RL-SCAP dynamically adjusts the slot allocation based on feedback from the environment. Rewards and penalties based on the success or failure of transmission efforts and the present status of the network as a whole drive the decision-making process of each node. Through this adaptive process, RL-SCAP can reduce collisions, increase the network's overall communication efficiency, and progressively improve its performance.

#### 4.1. Slot Allocation

RL-SCAP assigns time slots to IoT devices in a given sector using a reinforcement learning technique. As an RL agent, every node chooses the best time slot for its communication requirements by continuously learning from previous transmissions. To reduce interference and increase transmission reliability, the protocol makes sure that every slot is exclusively used by one sensor node.

Time is divided into fixed-length frames by RL-SCAP, and each frame has many time slots. In order to minimize the possibility of collisions with other devices in the network, each device independently determines the optimal time slot for data transmission. The selection procedure is based on Q-learning, in which each device modifies its strategy based on input it has observed, reducing the chance of having several transmissions during the same time slot.

#### 4.2. Autonomous Decision-Making Using Reinforcement Learning

Since collisions may happen only among nodes sharing the same channel, our RL-SCAP will be run independently on every sector. Accordingly, each sensor node in a given sector of angle  $\alpha$  functions as an RL agent that learns to choose the best time slot for its transmissions, according to the reinforcement learning paradigm that underpins the RL-SCAP slot allocation. Every device can observe its surroundings, make decisions depending on its current condition, and receive reward feedback that directs its learning process as described in Algorithm 1. The following are the primary steps in the RL-SCAP slot allocation procedure:

1. **State Observation:** Each device monitors its environment to gather information about the network, including the following:
  - Previous transmission failures or successes (deliveries or collisions). More specifically, each node needs to know which selected slots suffered collisions or hosted successful transmissions and which ones were not selected.
  - The level of congestion or interference observed in the previous frame. More precisely, each node needs only to know the maximum number of collisions experienced in selected slots that suffered interference. Our solution is designed to be scalable and can support up to 72,000 nodes by dividing the network into 360 sectors, with each sector handling up to 200 nodes. The proposed solution is not fully centralized. In a centralized setup, the gateway or server would run the entire RL algorithm and send the selected slots to the nodes. Instead, our

approach is partially decentralized. This design allows the nodes to participate in the decision-making process by observing the network conditions—particularly whether a collision has occurred—and then making informed choices. However, relying solely on a fully distributed solution is not suitable, since nodes only have access to their local views. This limited perspective may lead to incorrect slot selections, preventing the algorithm from converging. In our case, convergence means that every node eventually selects a mutually exclusive slot, with no collisions. To address this, we adopt a hybrid solution where nodes use their local observations to learn and make decisions, while the gateway supports them by sending a small vector containing the number of collisions per slot. This vector is lightweight and can be transmitted easily, without creating communication overhead. The gateway's guidance ensures that nodes make better-informed decisions, improving convergence and scalability.

2. **Selection of Actions:** Every device chooses an action (i.e., a particular time slot) in order to maximize the expected reward based on its current state. Actions are first chosen at random (exploration), but as the device gains experience, it starts to prioritize actions that will result in successful transmissions (exploitation).
3. **Reward Feedback:** The device is rewarded based on whether the transmission was successful or unsuccessful after every transmission. The device is penalized if the transmission causes a collision, and rewarded if the transfer is successful. More precisely, if the node transmission on a given slot is successful, a very high reward will be assigned in order to foster the selection of that slot. However, if the node experiences a collision on the selected slot, it has to change its slot. To do so, it will assign a reward to every slot based on the level of congestion on that slot. In particular, the node will assign a negative reward to the previously selected slot that hosts a successful transmission. By doing so, we guarantee that such a slot will not be selected by the node in order not to cause future collision. However, the higher the level of congestion in a given slot, the lower the assigned reward to that slot. Please note that slots that suffered collisions need to have inversely proportional positive reward in order to enable their selection in the upcoming transmissions. Indeed, assigning negative rewards to those slots will totally exclude them from the selection process and thus they will be totally unused. The device receives feedback from the reward, which directs its learning process. The device's Q-values, which show the anticipated long-term benefits of choosing specific slots, are updated using this feedback.
4. **Q-Value Update:** Every device tracks the anticipated rewards for every time slot it has access to in a Q-table. Following feedback, the device uses the following formula to update its Q-values:

The Q-value update rule is given by the following:

$$Q(s_t, a_t) = Q(s_t, a_t) + \beta \left( r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \quad (5)$$

where

- $Q(s_t, a_t)$  is the Q-value for the state  $s_t$  and action  $a_t$  (time slot choice) at time  $t$ .
- $r_t$  is the reward received after taking action  $a_t$ .
- $\beta$  is the learning rate (how much the new Q-value affects the old one).
- $\gamma$  is the discount factor (how much future rewards are valued).
- $\max_{a'} Q(s_{t+1}, a')$  is the maximum Q-value for the next state  $s_{t+1}$ .

By iteratively updating the Q-values, the device gradually improves its ability to select time slots that minimize collisions and maximize successful data transmissions.

This procedure enables nodes to continuously modify their communication techniques in response to the observed rewards, which eventually results in fewer collisions and more effective resource consumption.

---

**Algorithm 1:** Reinforcement Learning Slot Allocation Protocol (RL-SCAP)
 

---

**Input:** Number of slots (*nrofslots*), Number of channels (*nrofchannels*), Number of IoT devices (*nbr\_sensorAgent*)

**Output:** Optimized *slot\_vector* and *channel\_vector* for all IoT devices

**Initialize network parameters:** *nrofslots*, *nrofchannels*, *nbr\_sensorAgent*;

**Initialize learning parameters:** For each IoT device *i*, set channel ID, Q-table, learning rate ( $\alpha$ ), discount factor ( $\gamma$ ), and exploration rate ( $\epsilon$ );

**while** convergence criterion is not met **do**

**foreach** time interval **do**

        Observe the network state *s* (current slot allocation, collision feedback);

        Update the frame table (*ft*) for all IoT devices;

**foreach** IoT device *i* **do**

**if** slot and channel in *ft* cause collision **then**

                Update *collision\_vector*(*i*) based on detected collisions;

**else**

                Set *collision\_vector*(*i*) to 0;

**end**

**end**

**foreach** IoT device *i* **do**

            Select an action *a* (slot and channel) using  $\epsilon$ -greedy policy;

            Update *slot\_vector*(*i*) based on action *a*;

**end**

**foreach** IoT device *i* **do**

            Calculate reward *r* based on transmission success, collision, or idle time;

            Update  $Q(s, a)$  using:

$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ ;

**end**

**end**

**end**

**return** Optimized *slot\_vector*;

---

#### 4.3. Collision Detection and Mitigation

The main goal of the RL-SCAP protocol is to reduce collisions and maximize the distribution of mutually exclusive data transmission time slots. An adaptive reward assignment system makes this possible by enabling nodes to modify their transmission in real time in response to the results of prior transmission attempts. The protocol allows each node to gradually improve its behavior by giving feedback in the form of rewards or penalties, which lowers interference and increases network performance. A major objective of RL-SCAP is to reduce node collisions, which happen when two or more devices in the same sector try to send simultaneously on the same time slot. In a given sector, to keep track of each device's slot assignments and identify collisions, RL-SCAP employs a collision vector at the gateway that describes the level of congestion in every slot. This vector will be used by every node to autonomously assign rewards to every slot.

- **Reward Assignment:** The first step in the reward assignment process is to assess the collision status of the node's allocated time slot. The node receives a significant positive reward if there are no collisions. This reinforces effective behavior by encouraging the node to keep its selected time slot for subsequent broadcasts. On the other hand, the protocol imposes different penalties or lesser rewards based on the degree of congestion during the time window when a collision occurs.

- High congestion, which occurs when several nodes use the same slot for transmission, results in a low positive reward. Indeed, assigning a negative reward to that slot will result in excluding it from the selection process. Consequently, it will end up unused. That being said, assigning that slot a relatively low reward will enable its potential selection while being unfavorable.
- Moderate congestion, which occurs when fewer nodes use the same slot for transmission, results in a slight positive reward, indicating that, with prudence, the slot might still be available for future transmissions.
- No congestion, which occurs when a single node sends in the slot, results in a severe penalty to prevent the node from selecting that slot. Indeed, if a node transmission experiences a collision, it most probably has to change its slot. However, it does not have to switch to another slot that was previously selected by another node and hosted a successful transmission. By doing so, we prevent future collisions on that slot.
- A non-selected slot will be assigned a relatively high reward in order to favor its selection in the subsequent trial.

The following Algorithm 2 shows that how reward is assigned:

---

**Algorithm 2:** Reward Assignment Algorithm

---

```

Input: Node  $n$ , Time slot  $t$ 
Output: Assigned reward  $R(n, t)$ 
if node  $n$  has no collision at current time slot then
    | give  $n$  a reward of 1000 at current time slot;
end
else
    foreach time slot  $t$  do
        Find the maximum collision value for all nodes at  $t$ ;
        if  $value = 0$  then
            | give  $n$  a reward of  $-10,000$  at  $t$ ;
        end
        else if  $value = -3$  then
            | give  $n$  a reward of 10 at  $t$ ;
        end
        else if  $value = 1$  then
            | give  $n$  a reward of 5 at  $t$ ;
        end
        else if  $value = 2$  then
            | give  $n$  a reward of 3 at  $t$ ;
        end
        else if  $value = 3$  then
            | give  $n$  a reward of 1 at  $t$ ;
        end
        else
            | give  $n$  a reward of 0.5 at  $t$ ;
        end
    end
end

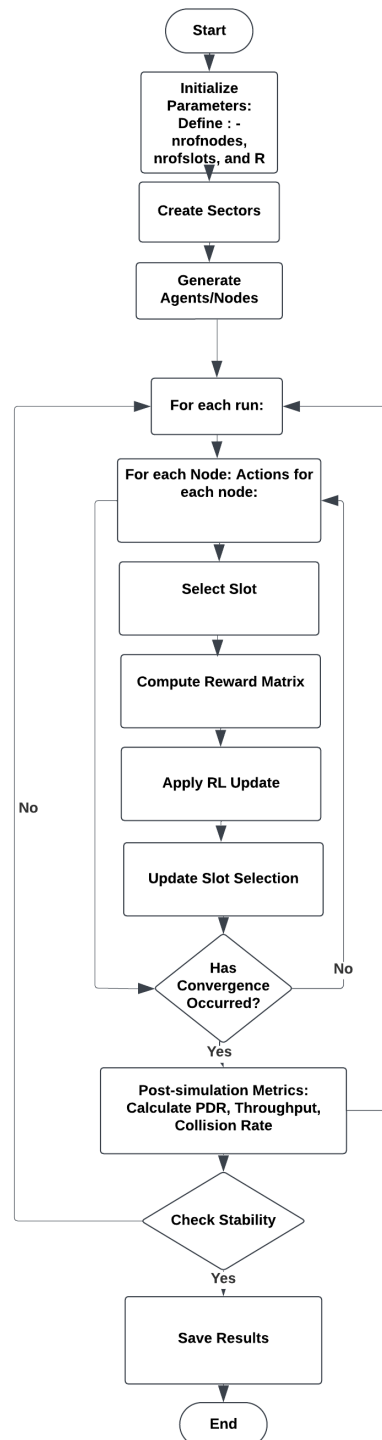
```

---

By learning from both successful and unsuccessful transmissions, nodes are able to continuously modify their slot selection technique thanks to this reward system.

Figure 4 illustrates the workflow of the RL-based slot allocation protocol's (RL-SCAP) mechanisms for a clearer understanding of how the protocol works. The process starts by initializing important parameters, such as the network radius ( $R$ ), number of slots (nrofslots), and number of nodes (nrofnodes). The system generates sectors and agents or nodes within them. After that, the process goes into a loop, where the simulation is performed several times. Each node performs actions in each episode of each run by choosing a slot, applying a

reinforcement learning update, calculating a reward matrix depending on performance, and modifying its slot selection accordingly. This iterative process continues until convergence is reached. Post-simulation parameters, including collision rate, throughput, and packet delivery ratio (PDR) are computed after convergence. After verifying that the results are stable, the system saves them if the stability requirements are satisfied. The workflow then comes to an end, monitoring stability and important performance parameters while using reinforcement learning to ensure an optimal slot allocation method.



**Figure 4.** The workflow of Reinforcement Learning-based slot- and channel-allocation protocol (RL-SCAP).



#### 4.4. Description of Algorithm

As shown in Algorithm 1 an organized process is used by the Reinforcement Learning Slot Allocation Protocol (RL-SCAP) to ensure a mutually exclusive slot assignment for Internet of Things devices. The first step is to define the network parameters, such as the number of slots, channels, and Internet of Things devices. Key reinforcement learning parameters are set by each IoT device, which initializes a Q-table to store learned policies. These parameters include the exploration rate ( $\epsilon$ ), which balances exploration and exploitation, the discount factor ( $\gamma$ ), which establishes the significance of future rewards, and the learning rate ( $\beta$ ), which regulates the amount that new information supersedes old values.

In order to stabilize slot allocation, the learning process is iterative and continues until convergence. The network state is monitored at every time step, recording the current slot allocation as well as collision feedback. These assignments are tracked using a frame table (ft). Then, every IoT device determines if the auto-assigned slot has experienced a collision. If there is a collision, the *collision\_vector(i)* is updated appropriately; if not, it is reset to zero.

Each IoT device uses a  $\epsilon$ -greedy policy to choose an action in order to identify the best slot assignment. By randomly choosing a slot with probability  $\epsilon$ , this policy permits exploration; however, exploitation takes place with probability  $1 - \epsilon$  by selecting the most well-known assignment from the Q-table. Each device's *slot\_vector(i)* is updated by the chosen action. After an action is performed, the result is assessed by a reward function as shown in Algorithm 2. A successful transmission carries a positive reward, whereas a collision carries a penalty or a relatively small reward.

Lastly, the typical Q-learning update rule is applied to update the Q-table is updated using reward Equation (5). Over time, the system can learn the best slot assignment thanks to this approach.

Table 1 shows symbols and their descriptions used in the RL-SCAP Protocol.

**Table 1.** Symbols and their descriptions used in the RL-SCAP Protocol.

Symbol	Description
<i>nrofslots</i>	Number of slots in the network.
<i>nrofchannels</i>	Number of orthogonal channels available.
<i>nbr_sensorAgent</i>	Total number of IoT devices in the network.
$Q(s, a)$	Q-value for state <i>s</i> and action <i>a</i> .
$\beta$	Learning rate for the reinforcement learning algorithm.
$\gamma$	Discount factor to prioritize immediate versus future rewards.
$\epsilon$	Exploration rate for the $\epsilon$ -greedy policy.
<i>r</i>	Reward value for a specific action taken by an IoT device.
<i>slot_vector</i>	Vector representing slot assignments for all IoT devices.
<i>channel_vector</i>	Vector representing channel assignments for all IoT devices.
<i>ft</i>	Frame table storing current slot and channel allocations.

## 5. Performance Evaluation

In this section, we evaluate the performance of our proposed protocol, RL-SCAP, and we compare it to SCAP SigFox in terms of collisions, the packet delivery ratio, and the network throughput. The MATLAB platform was chosen because of its strong machine learning and reinforcement learning (RL) capabilities. With built-in functions and toolboxes like the Reinforcement Learning Toolbox, which supports a variety of training strategies, policy optimization techniques, and deep learning integrations, MATLAB R2024b offers a complete environment for developing RL algorithms. Additionally, MATLAB is a good

option for creating and assessing the RL-SCAP protocol because of its capacity to effectively manage large-scale simulations, depict intricate system dynamics, and interface with other modeling tools. In this study, two different cases were examined to analyze the performance of the RL-SCAP protocol:

1. **Case 1: Fixed Number of Slots, Variable Number of Nodes:** In this case, the number of nodes grew while the number of time slots that were accessible remained fixed. This made it possible to test the RL-SCAP protocol in an environment of growing network congestion, where distributing limited resources (slots) among an increasing number of nodes was a challenge. The other reason for implementing such a scenario is that finding the number of needed slots is usually challenging. To deal with this, one of the solutions is to estimate the number of needed slots by estimating the number of nodes as derived in (3). However, estimating the number of needed slots  $m$  may result in either overestimation or underestimation in a real case scenario. Implementing such a scenario allows us to study both the overestimation case, when the number of nodes is less than the number of slots, and the underestimation case, when the number of nodes is greater than the number of available slots. With increasing network density, system performance was monitored in terms of collision rate, packet delivery ratio (PDR), throughput, and convergence time.
2. **Case 2: Equal Number of Slots and Nodes:** In this case, there were exactly the same amount of nodes and slots available. This configuration evaluated the effectiveness of the RL-SCAP protocol in allocating resources when the system's available slots and nodes are equal. The RL model's ability to sustain an ideal slot allocation approach in a setting where the number of available slots is equal to the number of nodes was examined.

Table 2 provides a summary of the main simulation parameters that we employed in our assessment.

**Table 2.** Parameters Settings.

Simulation Parameter	Fixed Value
Maximum Number of IoT Devices in a given sector	200
Number of channels per sector	1
Number of slots	Case 1 = 80, Case 2 = 20:20:200
Slot Duration	2000 ms (2 s)
Total Simulation Time	60,000 ms (1 min)
Number of Runs	800
Maximum Number of episodes	100
Bandwidth	100 Hz
$\beta$ (Learning Rate in RL)	0.1
Channel Size	100 Hz

Our solution is built upon the SCAP protocol, which relies on network partitioning to manage scalability and performance. The number of nodes we considered ranges from 20 to 200 within a single sector of the network. This choice aligns with the SCAP design principle, which divides the entire network into 360 sectors based on the angular division of a circle (each sector spans 1 degree). The selection of 20 to 200 nodes per sector reflects a realistic and scalable scenario. Indeed, since each sector operates independently, the total number of nodes in the entire network can be up to 72,000 (i.e., 200 nodes  $\times$  360 sectors). This level of scalability is made possible by Sigfox's ability to support 360 orthogonal channels, ensuring

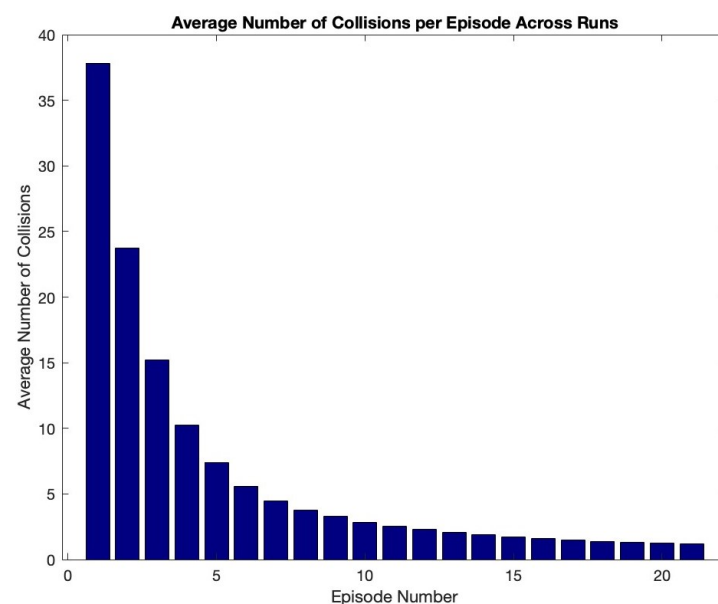
that nodes in different sectors do not interfere with each other. We emphasize that our reinforcement learning (RL) solution is applied locally within each sector. As such, even with a large number of total nodes, each RL instance only manages a limited, sector-specific subset (up to 200 nodes). This design guarantees numerical stability and computational feasibility, even when the overall network size grows substantially.

### 5.1. Evaluation of RL-SCAP Q-Learning Algorithm

To evaluate the speed of our Q-Learning algorithm, we examine the reduction in the average number of collisions over episodes as shown in Figure 5. A bar plot was created with an emphasis on the first 20 episodes. In this instance, we employed a fixed number of slots and nodes equal to 60.

The system's collision pattern indicates that the highest collision rates occur during the initial episodes, especially the first one. Nonetheless, there is a notable decrease in collisions throughout the first ten episodes. Following episode 30, the decrease becomes more gradual until it reaches minimal levels. After the initial episodes, the system shows quick stabilization in collision reduction with respect to convergence behavior. This pattern is consistent with the reinforcement learning (RL) process, in which the model moves from an exploration phase, which is marked by a high number of collisions, to an exploitation phase, which is when it distributes slots effectively and with few conflicts. Notably, there are 50 fewer collisions than there were in the first episode (about 130) compared with 80 in the second. This notable decrease in early episodes suggests that our RL model learns and optimizes slot allocation rapidly, hence reducing network interference to a large degree.

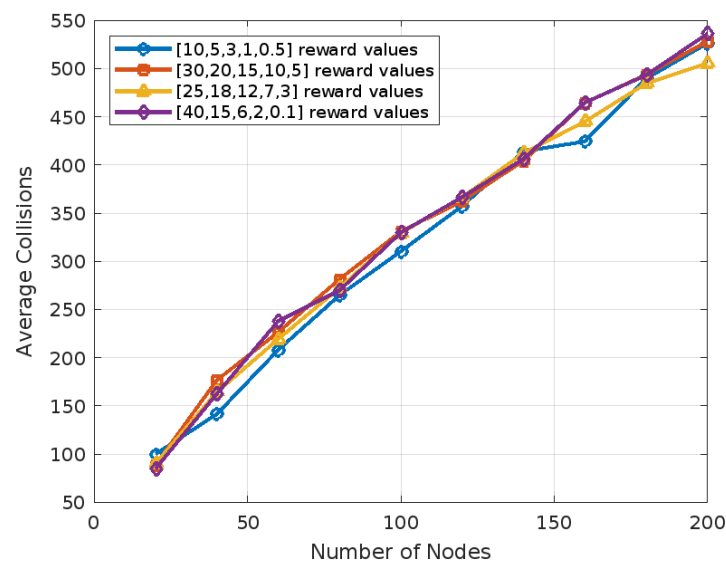
The initial contention is successfully handled by the RL-based method, indicating the protocol's flexibility and effective slot allocation as learning advances. This analysis demonstrates the ability of the RL-SCAP protocol to manage congestion and attain convergence, resulting in a notable decrease in collisions as episodes go.



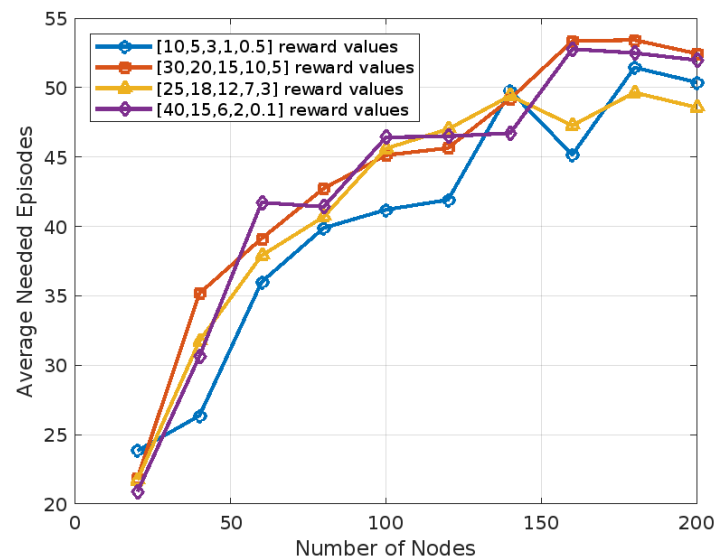
**Figure 5.** Average collisions per episode.

To evaluate the impact of different reward value sets on the performance of our reinforcement learning algorithm, we conducted a sensitivity analysis by testing several decreasing reward configurations. Specifically, we compared the baseline reward set with three alternative sets: [30, 20, 15, 10, 5], [25, 18, 12, 7, 3], and [40, 15, 6, 2, 0.1]. In each case, we examined the number of collisions and the number of episodes required for

convergence, as shown in Figures 6 and 7, respectively. Our findings show that the Q-learning algorithm is not highly sensitive to the exact values of the rewards, provided that the reward values decrease in magnitude. For instance, as shown in Figure 6, the percentage difference in the number of collisions compared with the original reward set was 7.22% for [30, 20, 15, 10, 5], 5.28% for [25, 18, 12, 7, 3], and 6.91% for [40, 15, 6, 2, 0.1], yielding an average deviation of 6.47%. A similar trend was observed in the number of episodes required to reach convergence. As shown in Figure 7, the percentage difference relative to the original set was 10.33% for [30, 20, 15, 10, 5], 7.18% for [25, 18, 12, 7, 3], and 10.01% for [40, 15, 6, 2, 0.1], with an average difference of 9.17%. These results confirm the robustness of our learning framework with respect to reward shaping, as long as the reward structure remains consistently decreasing.



**Figure 6.** Comparison of collisions in different reward value sets.



**Figure 7.** Comparison of needed episodes in different reward value sets.

### 5.2. Evaluation of RL-SCAP

This section evaluates our RL-SCAP protocol in two distinct scenarios to determine its performance:

- Fixed number of slots with a varying number of nodes scenario: There are 80 slots available, yet there are 20 to 200 nodes. The fixed number of slots will be estimated using Equation (3)
- Equal slots and nodes scenario: In this scenario, the number of slots is set equal to the number of nodes, which varies from 20 to 200 with an increment of 20.

We used the following performance criteria to assess the performance of RL-SCAP:

1. Number of Collisions until Convergence: The total number of collisions observed until the learning process stabilizes.
2. Number of needed episodes until Convergence: The number of episodes needed to achieve convergence in the RL-based slot allocation.
3. Packet Delivery Ratio (PDR): The ratio of successfully transmitted packets to the total packets sent, indicating communication reliability.

We used the following formula to calculate PDR:

$$PDR = \frac{\text{Total Successful Packets}}{\text{Total Sent Packets}} \quad (6)$$

4. Throughput: A measure of network efficiency that shows the total amount of data delivered successfully over time.

We used the following formula to calculate Throughput:

$$\text{Throughput} = \frac{\text{Total Successful Packets}}{nrofslots \times slotduration \times needed\_episodes} \quad (7)$$

#### 5.2.1. Scenario 1: Fixed Slots and Varying Number of Nodes

In this case, the number of nodes varies from 20 to 200, while the protocol performance is assessed with a fixed number of slots that equals 80.

As shown in Figure 8, two behaviors can be noted. The first one is when the number of nodes is less than or equal to the number of available slots (80), there is a gradual increase in the number of collisions and needed episodes until convergence is observed. The second behavior is experienced when the number of nodes is higher than the number of available slots. In this case, an important increase in the number of collisions is experienced between two consecutive values of the number of nodes, as convergence will never be reached. Indeed, when the number of nodes is higher than the number of available slots, mutually exclusive assignment of slots will never be reached; hence, the number of needed episodes is 100, which is the maximum set value in our simulations.

Regarding the PDR, as shown in Figure 9, a local minimum is observed at 60 nodes, while a global maximum occurs at 80 nodes. This behavior is driven, first, by the increasing number of collisions as more nodes compete for limited slots, and second, by the simultaneous rise in the number of transmissions, which temporarily enhances packet delivery. The trade-off between these two factors leads to the formation of local and global optima. After 80 nodes, both PDR and Throughput decline. For instance, when the number of nodes reaches 200 (which is 2.5 times the number of available slots), the PDR drops to approximately 0.88–0.89, while the Throughput stabilizes around 4.5 pps. This confirms that beyond 80 nodes, increased contention negatively impacts PDR, but Throughput remains steady, likely due to a balance between successful transmissions and collisions.



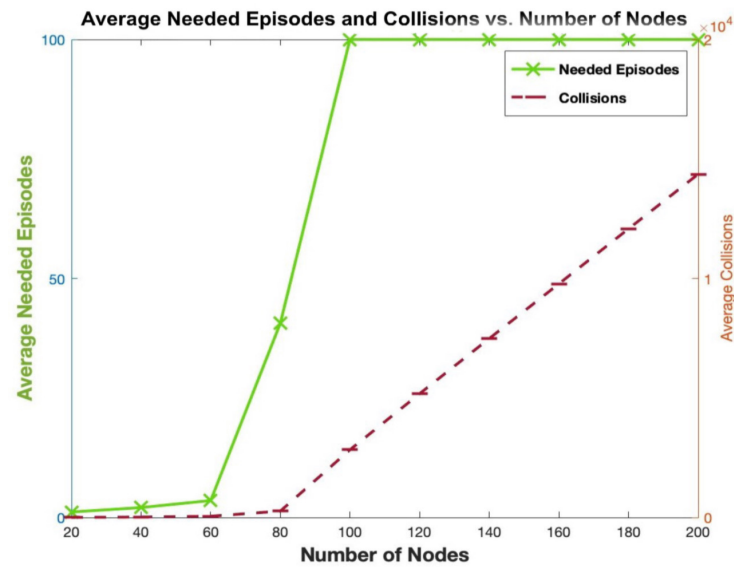


Figure 8. Case 1: Average Needed Episodes and Collisions.

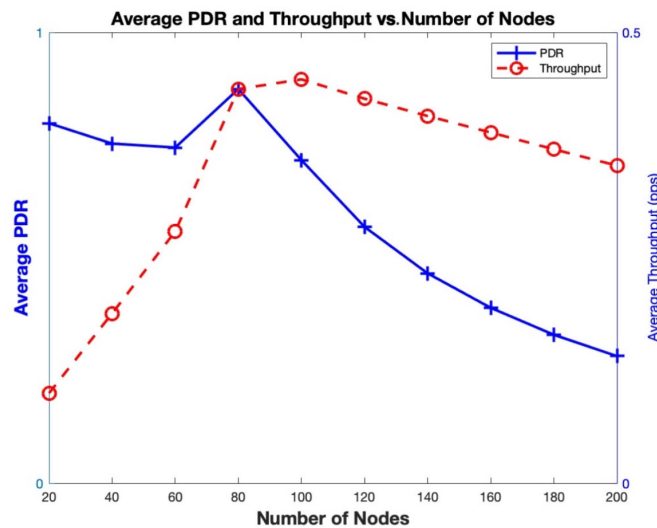


Figure 9. Case 1: Average PDR and Throughput.

#### 5.2.2. Scenario 2: Equal Slots and Nodes

This scenario assesses the performance of our RL-SCAP when the number of slots is set equal to the number of nodes. In this case, convergence is possible as the mutually exclusive slot assignment can be achieved.

As shown in Figure 10, when the number of nodes increases, as expected, the number of collisions until convergence increases, as well as the episodes needed to achieve the mutually exclusive slot assignment.

Figure 11 shows an increase in the throughput as well as the packet delivery ratio (PDR) when the number of nodes increases. Indeed, thanks to our adopted Q-learning algorithm, increasing the number of nodes will increase the number of successfully received packets in every episode faster than the increase in the number of collisions which confirms the scalability of our protocol. For instance, at 80 nodes,  $PDR \approx 0.89$ ,  $Throughput \approx 4.45$  pps. At 200 nodes,  $PDR \approx 0.91$  and  $Throughput \approx 4.55$  pps.

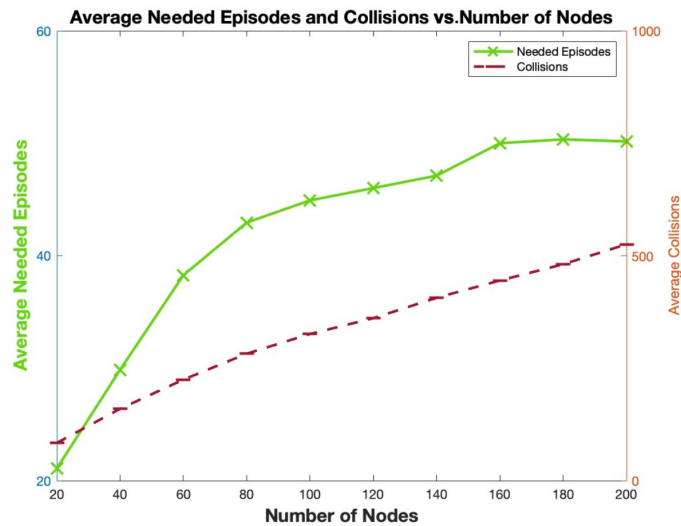


Figure 10. Case2: Average Needed Episodes and Collisions.

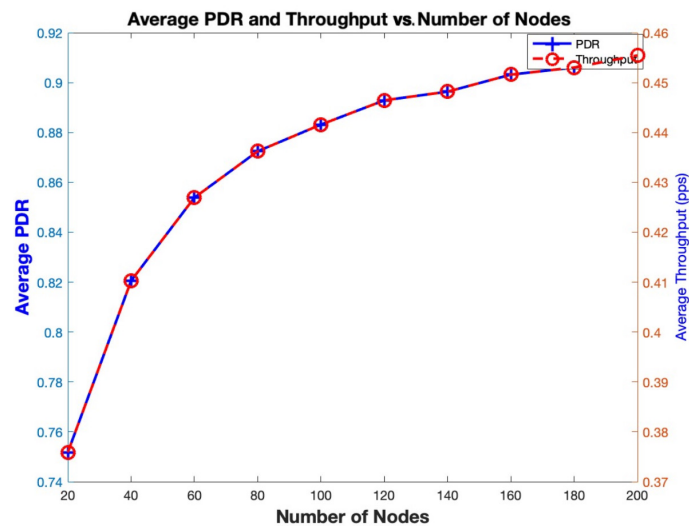


Figure 11. Case2: Average PDR and Throughput.

### 5.3. Comparison of RL-SCAP with SCAP

In this section, we compare the performance of RL-SCAP with SCAP SigFox. Recall that RL-SCAP was built upon SCAP SigFox, which highly outperforms the original SigFox protocol.

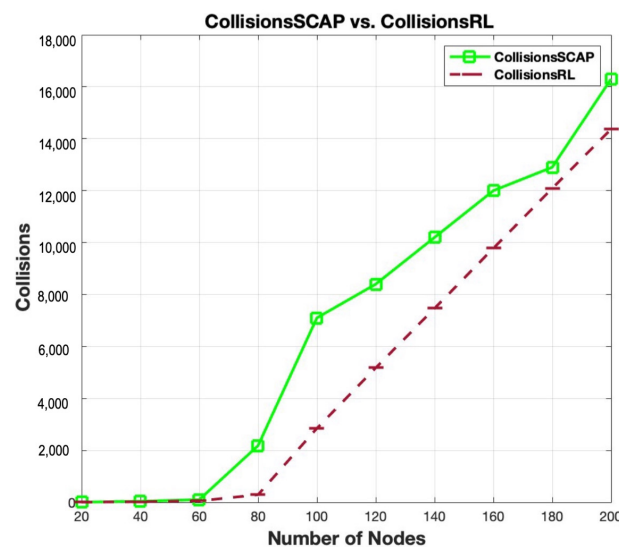
#### 5.3.1. Case 1 Scenario

In Figure 12, as the number of nodes increases, the number of collisions in SCAP and RL-SCAP rises. However, RL-SCAP regularly achieves lower collisions than SCAP, especially when the number of nodes is greater than or equal to 80, which is the number of available slots. Initially, both protocols encounter insignificant collisions when the number of nodes is small (less than 80), as the number of available slots is larger than the number of nodes. Indeed, with increased network density, more nodes tend to be within the same vicinity, which, according to SCAP, will end up selecting the same slot as they are within the same distance from the gateway, whereas RL-SCAP grows more steadily. Although both protocols achieve high collisions as the number of nodes is greater than the number of available slots, in greater node densities (above 100 nodes), the disparity in the number of collisions becomes more noticeable, suggesting that RL-SCAP is more effective in preventing collisions. By using

reinforcement learning, RL-SCAP improves slot allocation and decreases node contention, leading to better collision management and increased network efficiency. We can see that RL-SCAP reduces collisions by an average of 37.71%.

Packet Delivery Ratio (PDR) comparisons between the two protocols are shown in Figure 13. Although RL-SCAP PDR falls when the number of nodes exceeds the number of available slots, it still achieves higher values than the SCAP one, even when the number of nodes is less than the number of available slots. Indeed, since SCAP uses a distance-based slot assignment, as the number of nodes increases in a given sector of angle  $\alpha$ , the number of nodes that are within the same distance from the gateway will increase. Thus, the number of nodes selecting the same slot will increase, which will highly impact the ratio of successfully received packets to the total sent packets, since in SCAP, nodes sharing the same slot will always collide. For instance, when the number of nodes exceeds 40, SCAP's PDR drops precipitously, falling below 0.5 to reach less than 0.2 when the number of nodes equals 200. However, despite growing network congestion, RL-SCAP maintains a noticeably higher PDR across all node counts. The learning-based slot allocation appears to increase transmission success rates, as indicated by the peak PDR for RL-SCAP (0.85). Our improvement in PDR can reach up to 66.66%.

Better packet delivery performance is shown by RL-SCAP, indicating its capacity to efficiently handle network conflict. An improved PDR denotes greater use of resources, which makes the communication network more dependable.



**Figure 12.** Case 1: CollisionsSCAP vs. CollisionsRL.

Similar to the PDR, RL-SCAP achieves higher throughput than the SCAP protocol as shown in Figure 14. Despite its low achievement, the throughput of SCAP slowly increases until saturation is reached when the number of nodes equals 180, after which it starts decreasing. Indeed, in the first phase, when the number of nodes is less than 180, the SCAP network succeeds in delivering slightly more packets despite the high increase in collisions as the number of transmissions increases with the number of nodes. In fact, as the throughput is a compromise between the number of transmission attempts and the number of collisions, when the number of nodes is less than 180, the total number of transmissions is greater than the number of collisions. However, after saturation, the number of collisions highly dominates the number of transmission attempts, hence the decrease in throughput.

Despite the fact that RL-SCAP performs better by effectively allocating slots to balance transmissions and reduce collisions, SCAP's growing throughput is caused by the increased total number of transmissions. The dramatic increase in collisions, however, highlights

SCAP's inefficiency and eventually restricts its scalability. This distinction shows why RL-SCAP is a preferable method for dense networks because it generates a more stable and effective throughput curve. On average, RL-SCAP increases throughput by 39.12%.

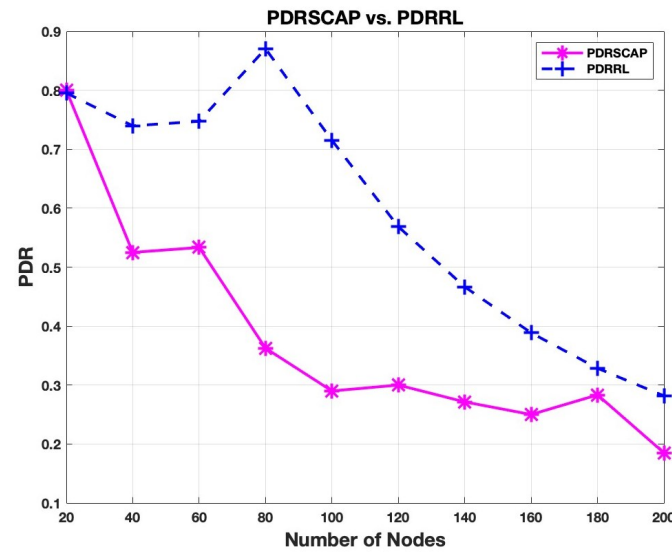


Figure 13. Case 1: PRDSCAP vs. PDRRL.

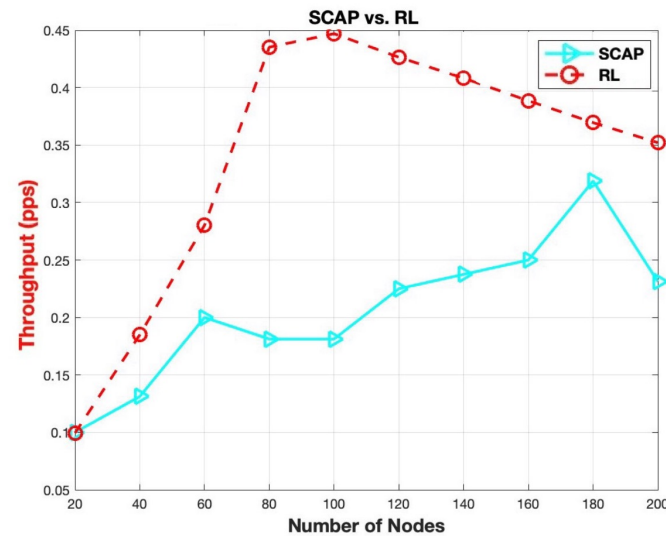


Figure 14. Case 1: ThroughputSCAP vs. ThroughputRL.

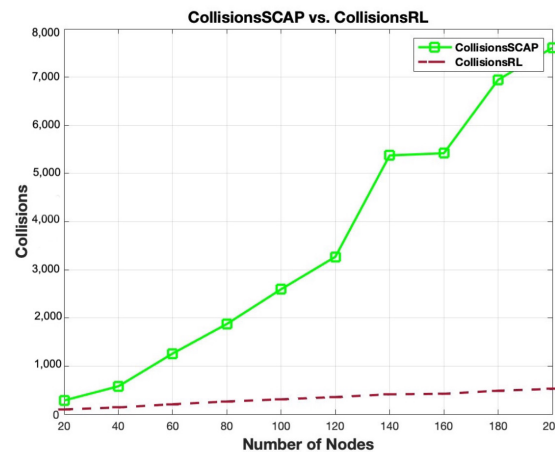
- Overall Findings:

1. Decreased Collisions: RL-SCAP effectively reduces collisions, improving network stability.
2. Increased PDR: By guaranteeing that more packets are delivered successfully, the reinforcement learning-based strategy increases communication dependability.
3. Increased Throughput: In situations involving dense networks, RL-SCAP performs better than SCAP in terms of data transmission efficiency.

### 5.3.2. Case 2 Scenario

Figure 15 compares the collisions between SCAP and RL-SCAP. As the number of nodes rises, SCAP shows a substantial rise in collisions. When the number of available slots equals the number of nodes, RL-SCAP consistently maintains noticeably lower collision rates. As the number of nodes increases, the difference between SCAP and RL-SCAP

becomes wider, suggesting that RL-SCAP scales more effectively. SCAP continues to have significant collision rates despite slots growing proportionally. This implies that the deterministic slot allocation method of SCAP is unable to adjust to the expansion of the network. In contrast, RL-SCAP uses reinforcement learning to allocate slots efficiently, lowering congestion and enhancing network stability. RL-SCAP decreases collisions by an average of 79.37%, with a highest reduction of 80.00%.



**Figure 15.** Case 2: CollisionsSCAP vs. CollisionsRL.

A comparison of the two protocols' PDR performance is shown in Figure 16. A consistently high PDR (0.75 to 0.95) is attained with RL-SCAP when the number of available slots equals the number of nodes, indicating efficient packet delivery. SCAP, in contrast, has lower PDR values, which range from 0.4 to ultimately falling as the number of nodes rises. RL-SCAP's PDR exhibits a steady trend as the number of nodes rises. Even as the network grows, packet delivery becomes more dependable thanks to RL-SCAP's adaptive slot allocation. On the other hand, SCAP's lower PDR results from ineffective scheduling and frequent collisions, which prevent many packets from successfully reaching the gateway. RL-SCAP boosts PDR by an average of 60.58%, with a maximum improvement of 74.47% when the number of available slots is set equal to the number of nodes.

The throughput of the two protocols when the number of nodes equals the number of available slots is depicted in Figure 17. As the number of nodes rises, RL-SCAP attains a significantly higher throughput than SCAP, reaching about 0.45 pps. As the number of nodes grows, SCAP's throughput decreases and stays low. Although the number of available slots equals the number of nodes, the deterministic distance-based slot assignment of SCAP poorly allocates the slots, as nodes within the same distance from the gateway will always share the same slots, leaving a number of available slots unused. Consequently, increasing the number of nodes in the same sector area will increase the number of nodes within the same distance from the gateway that will select the same slot despite the availability of unused slots. Thus, RL-SCAP is taking full advantage of the number of available slots as opposed to SCAP. The learning-based methodology of RL-SCAP dynamically adjusts to network conditions, guaranteeing effective utilization of available slots, whereas SCAP is unable to utilize the full available number of slots effectively. The average throughput improvement with RL-SCAP is 60.90%.

- Overall Findings

1. RL-SCAP Reduces Collisions More Effectively—SCAP still has significant collision rates despite the availability of more slots.
2. RL-SCAP Preserves a High PDR—SCAP's performance deteriorates with network growth, whereas the RL-based method guarantees reliable packet delivery.



3. RL-SCAP Achieves Higher Throughput: By making the most use of the resources at hand, RL-SCAP outperforms SCAP in terms of throughput.

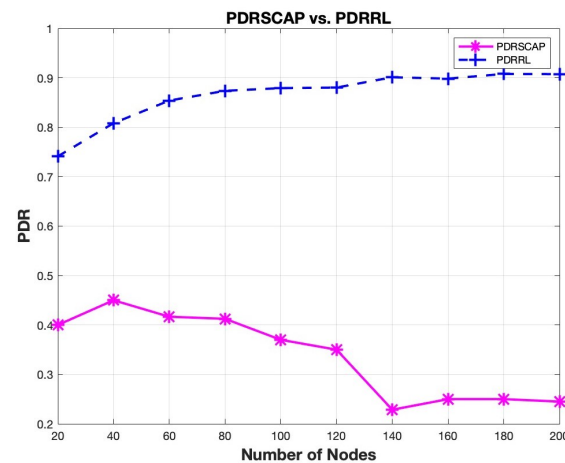


Figure 16. Case 2: PDRSCAP vs. PDRRL.

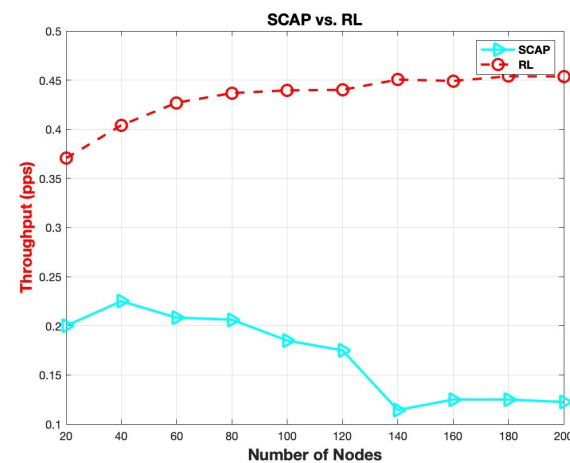


Figure 17. Case 2: Comparison of Throughput in RL-SCAP and SCAP.

#### 5.4. Summary of SCAP vs. RL-SCAP Performance

In order to demonstrate the performance enhancements that RL-SCAP achieves over SCAP, we present a side-by-side comparison of important performance metrics in both cases. As seen in Table 3, RL-SCAP performs significantly better than SCAP in terms of collision reduction, throughput, and packet delivery ratio (PDR).

Table 3. Comparative Summary of SCAP and RL-SCAP Performance.

Protocol	Scenario	PDR (%)	Throughput (pps)	Avg. Collisions
SCAP	Case 1: Fixed Slots	~20–60	~0.2–4.5	~60–180
RL-SCAP	Case 1: Fixed Slots	~65–85	~3.5–8.5	~30–90
SCAP	Case 2: Equal Nodes/Slots	~40–75	~2.0–6.5	~50–130
RL-SCAP	Case 2: Equal Nodes/Slots	~75–95	~4.0–10.5	~10–40

## 6. Conclusions

In this study, we introduced RL-SCAP, an extension of the SCAP [5] protocol based on reinforcement learning, which is intended to increase slot allocation for massive IoT networks. In order to achieve effective resource usage and minimize collisions, our protocol dynamically adjusts to network conditions by utilizing reinforcement learning. We showed through an extensive performance test that RL-SCAP outperformed classical SCAP in terms of important metrics, including collision rate, throughput, and packet delivery

ratio (PDR). According to our findings, RL-SCAP, on average, lowers collisions by 58.54%. Furthermore, RL-SCAP improves packet delivery performance significantly; the average PDR improvement was 53.82%. Additionally, network performance is increased by RL-SCAP, with an average improvement of 50.01%. These enhancements demonstrated that RL-SCAP could scale and manage high node densities without sacrificing energy efficiency. With this work, we improved network scalability and efficiency while preserving energy efficiency by optimizing slot allocation in low-power IoT networks. Because of these enhancements, RL-SCAP is especially well suited for applications, such as environmental monitoring, smart cities, and industrial IoT.

While RL-SCAP demonstrates promising results in collision reduction and throughput improvement, several limitations remain. First, the current evaluation is based on simulations with static and homogeneous node distributions, which may not fully capture the dynamics of real-world IoT deployments involving node mobility, heterogeneous traffic loads, or varying transmission priorities. Second, the protocol assumes synchronized time slots and accurate sectorization, which may require additional infrastructure in practical deployments. Finally, energy consumption overhead from continuous learning and slot monitoring may impact battery-constrained devices, especially in large-scale deployments. Future work will address these aspects by extending the protocol to handle adaptive reward tuning, real-world deployment constraints, and mobility-aware decision-making. Future research will focus on transitioning RL-SCAP from simulation to real-world deployment and evaluating its effectiveness in practical IoT environments. Please note that the transition from simulation to real deployment is relatively straightforward in our case due to the protocol's lightweight design and decentralized decision-making. Each node can locally determine its sector assignment based on its known geographic coordinates and those of the gateway, exactly as conducted in SCAP. This spatial sectoring step does not require complex computation or centralized control. Once the sector is identified, the RL-based slot selection operates as described, using local observations (i.e., collision feedback) and lightweight assistance from the gateway. This will help uncover real-world implementation challenges related to energy consumption, deployment scalability, and hardware constraints. Additionally, further research will explore adaptive energy management techniques to extend battery life while maintaining high network efficiency. To further enhance RL-SCAP's performance in large-scale deployments, future work could focus on refining RL-based decision-making at the gateway, leveraging historical network conditions to optimize slot allocation more effectively.

**Author Contributions:** Conceptualization, R.A. and F.B.; methodology, R.A. and F.B.; software, R.A.; validation, R.A., F.B., L.M. and S.S.; formal analysis, R.A.; investigation, R.A.; resources, R.A.; data curation, R.A.; writing—original draft preparation, R.A.; writing—review and editing, R.A., F.B., L.M. and S.S.; visualization, R.A.; supervision, F.B., L.M. and S.S.; project administration, F.B., L.M. and S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available in the article.

**Acknowledgments:** The authors would like to sincerely thank the Department of Information Technology, Faculty of Computing and Information Technology, for offering a helpful study environment. The first author would especially want to express her gratitude to Fatma Bouabdallah, for her exceptional guidance, continuous support, and invaluable insights throughout this research. Additionally, Linda Mohaisen and Shireen Saifuddin are appreciated for their supportive comments and encouragement during the creation of this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Chaudhari, B.S.; Zennaro, M.; Borkar, S. LPWAN technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet* **2020**, *12*, 46. [\[CrossRef\]](#)
2. Chilamkurthy, N.S.; Pandey, O.J.; Ghosh, A.; Cenkeramaddi, L.R.; Dai, H.N. Low-power wide-area networks: A broad overview of its different aspects. *IEEE Access* **2022**, *10*, 81926–81959. [\[CrossRef\]](#)
3. Raza, U.; Kulkarni, P.; Sooriyabandara, M. Low power wide area networks: An overview. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 855–873. [\[CrossRef\]](#)
4. Lavric, A.; Petrariu, A.I.; Popa, V. Sigfox communication protocol: The new era of IoT? In Proceedings of the 2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI), Lisbon, Portugal, 29–30 August 2019; IEEE: New York, NY, USA, 2019; pp. 1–4.
5. Alqurashi, H.; Bouabdallah, F.; Khairullah, E. SCAP SigFox: A Scalable Communication Protocol for Low-Power Wide-Area IoT Networks. *Sensors* **2023**, *23*, 3732. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Reynders, B.; Wang, Q.; Tuset-Peiro, P.; Vilajosana, X.; Pollin, S. Improving Reliability and Scalability of LoRaWANs through Lightweight Scheduling. *IEEE Internet Things J.* **2018**, *5*, 1830–1842. [\[CrossRef\]](#)
7. Saluja, D.; Singh, R.; Baghel, L.K.; Kumar, S. Scalability Analysis of LoRa Network for SNR-Based SF Allocation Scheme. *IEEE Trans. Ind. Inform.* **2020**, *17*, 6709–6719. [\[CrossRef\]](#)
8. Adame, T.; Bel, A.; Bellalta, B. Increasing LPWAN Scalability by Means of Concurrent Multiband IoT Technologies: An Industry 4.0 Use Case. *IEEE Access* **2019**, *7*, 46990–47010. [\[CrossRef\]](#)
9. Shen, H.-H.; Wu, C.-H.; Su, W.-C.; Feng, K.-T. Analysis and Implementation for Traffic-Aware Channel Assignment and Contention Scheme in LoRa-Based IoT Networks. *IEEE Internet Things J.* **2021**, *8*, 11368–11383. [\[CrossRef\]](#)
10. Osman, N.I.; Abbas, E.B. Simulation and Modelling of LoRa and Sigfox Low Power Wide Area Network Technologies. In Proceedings of the 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, 12–14 August 2018; IEEE: New York, NY, USA, 2018; pp. 1–5. [\[CrossRef\]](#)
11. Lavric, A.; Petrariu, A.I.; Popa, V. Long Range Sigfox Communication Protocol Scalability Analysis under Large-Scale, High-Density Conditions. *IEEE Access* **2019**, *7*, 35816–35825. [\[CrossRef\]](#)
12. Alahmadi, H.; Bouabdallah, F.; Al-Dubai, A.; Ghaleb, B. A Novel Autonomous Time-Slotted LoRa MAC Protocol with Adaptive Frame Sizes. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 917–922. [\[CrossRef\]](#)
13. Alahmadi, H.; Bouabdallah, F.; Al-Dubai, A.; Ghaleb, B. A Novel Time-Slotted LoRa MAC Protocol for Scalable IoT Networks. *Future Gener. Comput. Syst.* **2022**, *134*, 287–302. [\[CrossRef\]](#)
14. Barrachina-Muñoz, S.; Adame, T.; Bel, A.; Bellalta, B. Towards energy efficient LPWANs through learning-based multi-hop routing. In Proceedings of the 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019; IEEE: New York, NY, USA, 2019; pp. 644–649.
15. Yazid, Y.; Guerrero-González, A.; Ez-Zazi, I.; El Oualkadi, A.; Arioua, M. A Reinforcement Learning Based Transmission Parameter Selection and Energy Management for Long Range Internet of Things. *Sensors* **2022**, *22*, 5662. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Onishi, T.; Li, A.; Kim, S.-J.; Hasegawa, M. A Reinforcement Learning-Based Collision Avoidance Mechanism to Superposed LoRa Signals in Distributed Massive IoT Systems. *IEICE Commun. Express* **2021**, *10*, 289–294. [\[CrossRef\]](#)
17. Park, G.; Lee, W.; Joe, I. Network Resource Optimization with Reinforcement Learning for Low Power Wide Area Networks. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 176. [\[CrossRef\]](#)
18. Huang, X.; Jiang, J.; Yang, S.-H.; Ding, Y. A Reinforcement Learning Based Medium Access Control Method for LoRa Networks. In Proceedings of the 2020 IEEE International Conference on Networking, Sensing and Control (ICNSC), Nanjing, China, 30 October–2 November 2020; IEEE: New York, NY, USA, 2020; pp. 1–6. [\[CrossRef\]](#)
19. Zhong, H.; Ning, L.; Wang, J.; Suo, S.; Chen, L. Optimization of LoRa SF Allocation Based on Deep Reinforcement Learning. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1–14. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.