



**HAL**  
open science

# Heuristic-Deep Q-Network-based Network Slicing in LoRaWAN

Fatima Zahra Mardi, Miloud Bagaa, Yassine Hadjadj-Aoul, Nabil Benamar

► **To cite this version:**

Fatima Zahra Mardi, Miloud Bagaa, Yassine Hadjadj-Aoul, Nabil Benamar. Heuristic-Deep Q-Network-based Network Slicing in LoRaWAN. ICC 2023 - IEEE International Conference on Communications, May 2023, Rome, Italy. pp.4731-4736, 10.1109/ICC45041.2023.10278565 . hal-04368564

**HAL Id: hal-04368564**

**<https://inria.hal.science/hal-04368564v1>**

Submitted on 1 Jan 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Heuristic-Deep Q-Network-based Network Slicing in LoRaWAN

Fatima Zahra Mardi<sup>1</sup>, Miloud Bagaa<sup>2</sup>, Yassine Hadjadj-Aoul<sup>3</sup>, and Nabil Benamar<sup>1,4</sup>

<sup>1</sup>Moulay Ismail University, Meknes, Morocco. <sup>2</sup> Department of Electrical and Computer

Engineering, Université du Québec à Trois-Rivières, Trois-Rivières, QC, Canada.

<sup>3</sup>University of Rennes, Inria, IRISA, France. <sup>4</sup>Al Akhawayn University in Ifrane, Morocco.

fa.mardi@edu.umi.ac.ma, bagmoul@gmail.com, yassine.hadjadj-aoul@irisa.fr, n.benamar@umi.ac.ma

**Abstract**—Due to the increase in the number of Internet of Things (IoT) devices in recent years, managing and supporting the diversity of services is becoming more difficult. Network Slicing will be the solution, in which the network slices are tailored to the requirements of the services. In this paper, network slicing is investigated in LoRaWAN networks using the Heuristic-Deep Q-Network (H-DQN) solution that manages the network resource allocation. We propose an intra-service allocation based on the deep Q-Network (DQN) algorithm by allocating virtual resource blocks to services. In addition, the intra-service allocation is based on a heuristic algorithm that assigns the transmission probability to the LoRa nodes of each service for each block in a way to maximizes the Packet Delivery Rate (PDR) of the network while ensuring that the priority of services is maintained. Simulation results show that the proposed approach improves the PDR, and ensures prioritization among services.

**Index Terms**—Network Slicing, LoRaWAN network, Resource allocation, deep Q-Network (DQN).

## I. INTRODUCTION

With the growth of the Internet of Things (IoT) technology, billions of IoT devices are connecting and communicating through standard communication protocols to accommodate networks with varying application areas [1]. IoT applications require an increasing number of end devices that offer low power consumption, low-cost communications, high coverage, and long-range communications [2]. To address these challenges, Low-Power Wide Area Network (LPWAN) technology has been proposed as a new set of technologies for the IoT to achieve the objectives above at once. They promise, indeed, convenient connectivity for large-scale IoT deployments.

The LoRa Wide Area Network (LoRaWAN) radio communication technology is based on LoRa, which is a spread spectrum modulation technique derived from the Chirp Spread Spectrum (CSS) technology to communicate over long distances [3]. The CSS modulation in LoRa converts each data symbol into a chirp, which is a signal whose frequency linearly increases or decreases over time [4]. LoRaWAN is an open-source standard LPWAN protocol developed by LoRa Alliance [5]. It defines the Medium Access Control (MAC) layer specifications that build on the LoRa physical layer. In addition, the LoRaWAN network architecture has a star topology, which can support many IoT devices that can send data to the server through the gateways.

As for the MAC layer, every IoT device adopts aloha as an access protocol, whereby multiple users attempt to send data simultaneously over the same channel, which leads to increased collisions. Meanwhile, due to the increase in the number of IoT devices belonging to various applications with varying service requirements, the amount of traffic keeps rising, making it difficult to ensure even the slightest guarantee of the needs of the multiple applications.

To overcome these challenges, Network Slicing (NS) has emerged as a technological solution to support different services on the same shared physical layer through wireless resource allocation mechanisms that can be divided to meet the specific service requirements for each slice.

In our approach, we apply network slicing to the LoRaWAN transmission protocol. Therefore, this study's primary research issue is finding an allocation technique that improves the network performance and ensures the prioritization between services. The main contributions of this paper are:

- We propose a much finer allocation of resources by subdividing channels into time slots of fixed duration, allowing IoT LoRa nodes to be orchestrated on the resulting resources' blocks.
- We propose an inter-service allocation based on a DQN algorithm to allocate resources' blocks to services.
- According to the DQN allocation, we propose an intra-service allocation based on a heuristic algorithm to assign access probabilities to the LoRa nodes belonging to the services.

The remainder of the paper is organized as follows. In Section II, we present the related work. In Section III, we discuss the system model. Section IV presents the network slicing problem. Section V presents the proposed H-DQN solution. Section VI demonstrates the performance of the proposed solution and discusses the obtained results. Finally, section VII concludes the paper.

## II. RELATED WORK

Due to the many benefits of the network slicing paradigm for improving overall system performance and meeting network quality of service standards, its practical deployment has been the topic of several evaluations.

As network slicing is a concept emerging from cellular mobile networks, its application was first focused on use cases

in these networks. The author in [6] suggested the use of network slicing in LTE Networks by proposing a network slicing resource management (NSRM) system to distribute the required resources to each slice while considering resources' isolation between multiple slices. The authors in [7] investigate the benefits of non-orthogonal sharing of RAN resources to facilitate uplink communications between a set of eMBB, mMTC, and URLLC devices and a single base station.

More recently, some works have started to explore the possibility of applying this concept in LPWANs and in particular LoRaWAN networks. The authors in [8] suggested a framework for Industry 4.0, which is based on LoRaWAN and software-defined networking by proposing an online Gaussian mixture model (OGMMC) and dynamic mini-batch gradient descent (MBGD) algorithms to provide the slice allocation to the available channels. Meanwhile, the authors in [9] have proposed a channel allocation in LoRaWAN network employing the maximum likelihood estimation (MLE) method to estimate the resources' requirements of the different slices. Using the above estimate, the authors were able to derive a new strategy for resources allocation avoiding slice starvation while meeting the QoS requirement of the slices. The authors in [10] have proposed a centralized network slicing solution in LoRaWAN relying on a coalitional game. The initial coalitions were formed using the K-Means algorithm, and the strategy consisted in moving LoRa nodes from one coalition to another in a way to increase network performance while guaranteeing the appropriate SLA for each network slice. However, the latter approach has a high convergence latency that may not be compatible with networks where changes are relatively frequent.

As the data increases, the resource allocation faces more and more difficulties to satisfy the demands of the slices. With this in mind, researchers are increasingly resorting to deep reinforcement learning. The authors in [11] have leveraged Deep Federated Q-Learning (DFQL) strategy in LoRaWAN network for the channels' assignment based on the Transmission Power (TP) and the Spreading Factor (SF) adjustments and previous agents' experiences. The aim is to improve the QoS rewards of each slice members. However, resource allocation based on channels is very limited. In fact, in such a system, we allocate to a slice one or more channels, which means that we are quickly confronted with the limit of the number of channels. Indeed, as the number of services increases, it will be complicated to satisfy the demands of the supported services. By adding the time dimension, the resources become more numerous, and the allocation of the latter can be very tight.

In contrast to the previous solutions, in this paper, we propose a new approach to ensure fair resource allocation relying on the temporal subdivision of channels in a LoRaWAN network. We have also used H-DQN algorithm to improve the Packet Delivery Rate (PDR) while ensuring the priority between different services.

### III. SYSTEM MODEL

#### A. Network infrastructure

Without loss of generality, we consider, in the following, a scenario for LoRaWAN network slicing, consisting of one LoRa gateway GW, a set of Lora nodes  $\mathbb{N} = \{1, \dots, n\}$  randomly distributed in the communication area, within the radius of this gateway, the whole forming a star topology.

We denote by  $\mathbb{B}$  the total Resource Blocks (RBs) in the network based on time and frequency,  $\mathbb{C} = \{1, \dots, c\}$  denote the set of resource channels available at the gateway in the frequency dimension, and  $\mathbb{T} = \{1, \dots, t\}$  denote the set of time slots in the time dimension. Hence, we define each block as  $\{b_{ij} : i \in \mathbb{T}, j \in \mathbb{C}\}$ , we assume that a given block  $b_{ij} \in \mathbb{B}$  is assigned to only one service  $s \in \mathbb{S}$ , where  $\mathbb{S}$  denote the set of services.

$$\forall m, n \in \mathbb{S}, \mathbb{B}_m \cap \mathbb{B}_n = \emptyset. \quad (1)$$

where  $\mathbb{B}_k$  denotes the set of blocks assigned to the service  $k \in \mathbb{S}$ .

The slicing framework consists of splitting the total resource blocks into several virtual networks (slices) by allocating blocks to different slices of the network. For this purpose, we consider that each slice is associated with a service, let denote by  $\mathbb{S} = \{1, \dots, s\}$  the set of supported services. We consider that a given LoRa node  $n \in \mathbb{N}$  belongs to one service, and  $\mathbb{N}_s$  denote the set of LoRa nodes within a service  $s \in \mathbb{S}$ .

$$\forall k, l \in \mathbb{S}, \mathbb{N}_k \cap \mathbb{N}_l = \emptyset. \quad (2)$$

$$\mathbb{N} = \left| \bigcup_{k \in \mathbb{S}} \mathbb{N}_k \right| \quad (3)$$

#### B. Communication model

As a channel, We consider the log distance path model with shadowing [12]. The path loss is based on the distance  $d$  between the LoRa node  $n$  and the gateway (GW), which is given by Equation 4. For the shadowing, we assumed a normal distribution with zero mean and  $\sigma^2$  variance  $X_\sigma \sim N(0, \sigma^2)$  [12]

$$L_{pl,n}(d) = L_{pl}(d_0) + 10\Gamma \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (4)$$

where  $L_{pl}(d_0)$  is the mean path loss at the reference distance  $d_0$  in dB, and  $\Gamma$  is the path-loss factor.

The received signal power  $P_{rx}$  from the LoRa node  $n$  at the gateway, is described in Eq. 5 [12].

$$P_{rx} = P_{tx,n} + G - L - L_{pl,n} \quad (5)$$

where  $P_{tx,n}$  is the LoRa node's transmit power,  $G$  the antenna power gains,  $L$  the power losses at the transmitter, and  $L_{pl,n}$  is the path loss.

In order to guarantee a successful packet delivery from LoRa node  $n$  to the GW, the power  $P_{rx}$  must be larger than the receiver sensitivity  $S_{rx}$  for a specific Spreading Factor (SF) [12]. On the other hand, the primary cause of

packet loss is collision, which occurs when at least two LoRa nodes attempt to transmit packets simultaneously. In fact, the collision is impacted by a variety of factors including carrier frequency (CF), spreading factor (SF), transmission power, and time [12]. The main objective here is to maximize the service's PDR by ensuring priority between competing services.

#### IV. PROBLEM FORMULATION

The objective of this paper is based on: 1) Inter-service allocation, which consists in assigning the total Resource Blocks (RBs) to the services. 2) Intra-service allocation, which consists in assigning a transmission probability to LoRa nodes within a service for each block  $b_{ij}$  in order to maximize the PDR of the services, while ensuring prioritization between them.

Without loss of generality, we consider, in the following, three services, that differ in term of priority. The priority of the services is indicated by the desirable level of PDR for each service. Since the first service is the highest priority, it should have the highest PDR compared to the other services. The second service, on the other hand, has a higher priority than the third one, while the third one has the lowest priority and should have the lowest PDR rate.

To ensure the priorities of the services, the first step consists in allocating the intra-service resources in an optimized way by assigning the RBs between the services. A second step consists in assigning inter-service resources through the assignment of transmission probabilities to LoRa nodes taking into account the respect of priorities between the services with the objective of maximizing all the services' PDR.

Let  $\mathcal{X}_{b_{ij},s} \in \{0,1\}$  a boolean decision variable, which is equal to 1 if a block  $b_{ij} \in \mathbb{B}$  is used by a service  $s \in \mathbb{S}$ , otherwise, it equals to 0. Let denote by  $\mathcal{Y}_{n,b_{ij}}$  a continuous variable that shows the probability of a LoRa node  $n \in \mathbb{N}_s$  of a service  $s \in \mathbb{S}$  to transmit in the block  $b_{ij}$ .

As stated before, the objective is to maximize the PDR adopted by each service. This can be formulated as follow:

$$PDR_s = \sum_{b_{ij} \in \mathbb{B}} \mathcal{X}_{b_{ij},s} \times PDR_{b_{ij},s}, \forall s \in \mathbb{S} \quad (6)$$

, whereby  $PDR_{b_{ij},s}$  is the PDR of a service in a specific block  $b_{ij}$

$$PDR_{b_{ij},s} = \frac{\sum_{n \in \mathbb{N}_s} \mathcal{Y}_{n,b_{ij}} N_n^s - \sum_{n \in \mathbb{N}_s} N_n^c}{\sum_{n \in \mathbb{N}_s} \mathcal{Y}_{n,b_{ij}} N_n^s} \quad (7)$$

, whereby  $N_n^s$  denotes the number of packets sent by each LoRa node  $n \in \mathbb{N}_s$  in the block  $b_{ij}$  and  $N_n^c$  is the total number of colliding packets. To this end, the assignment optimization problem is formulated as follows:

$$\max_{\forall s \in \mathbb{S}} U_s = PDR_s \times w_s$$

subject to

$$\forall s \in \mathbb{S}, \forall n \in \mathbb{N}_s, \forall b_{ij} \in \mathbb{B} : \mathcal{Y}_{n,b_{ij}} \leq \mathcal{X}_{b_{ij},s} \quad (8a)$$

$$\forall b_{ij} \in \mathbb{B} : \sum_{s \in \mathbb{S}} \mathcal{X}_{b_{ij},s} = 1 \quad (8b)$$

$$\forall j, j' \in \mathbb{C}, \forall i \in \mathbb{T}, \forall s \in \mathbb{S}, \mathbb{N}_{s,b_{ij}} \cap \mathbb{N}_{s,b_{ij}'} = \emptyset. \quad (8c)$$

$$\forall s \in \mathbb{S}, \forall n \in \mathbb{N}_s, \forall b_{ij} \in \mathbb{B} : \mathcal{Y}_{n,b_{ij}} \leq 1 \quad (8d)$$

$$\forall s \in \mathbb{S}, \forall n \in \mathbb{N}_s, \forall b_{ij} \in \mathbb{B} : \mathcal{Y}_{n,b_{ij}} \geq 0 \quad (8e)$$

$$\forall b_{ij} \in \mathbb{B} : \sum_{s \in \mathbb{S}} \sum_{n \in \mathbb{N}_s} \mathcal{Y}_{n,b_{ij}} = 1 \quad (8f)$$

$$\forall s \in \mathbb{S}, \forall n_1, n_2 \in \mathbb{N}_s \wedge n_1 \neq n_2 : \sum_{b_{ij} \in \mathbb{B}} \mathcal{Y}_{n_1,b_{ij}} - \sum_{b_{ij} \in \mathbb{B}} \mathcal{Y}_{n_2,b_{ij}} \leq \epsilon \quad (8g)$$

, whereby  $w_s$  is a weight factor for each service.

Constraint (8a) ensures that a LoRa node can only use a block if it is already assigned to its service. The constraint (8b) ensures that each block must be assigned to one and only one service. Furthermore, (8c) ensure that if a service  $s \in \mathbb{S}$  is assigned to the blocks sharing the same  $i \in \mathbb{T}$  in the time domain, LoRa nodes belonging to this service cannot be shared between two channels  $j, j' \in \mathbb{C}$  in the frequency domain, i.e., the LoRa nodes in block  $b_{ij}$  are not the same as those in the block  $b_{ij}'$ . The constraints (8d) (8e) ensure the range of probability values that cannot be less than 0 or greater than 1. The probability allocation constraint (8f) implies that for each block assigned to a service, the sum of probability of the LoRa nodes in that service is equal to 1, while constraint (8g) ensures the equiprobability between LoRa nodes that belong to the same service.

The problem described above is a Mixed Integer Linear Program (MILP), which is by definition NP-hard. It can therefore only be solved for small problem instances. Therefore, we propose a suboptimal solution in the next section to deal with large network instances.

#### V. PROPOSED SOLUTION

In this section, we propose a resource allocation solution to maximize the PDR while guaranteeing priority among services in two steps. In the first step, we dynamically allocate resource blocks to services depending on their weights based on a Deep Q-Network (DQN) algorithm. In the second step, we allocate the transmission probability for each LoRa node on each service based on a heuristic algorithm. Fig. 1 illustrates the proposed framework for resource allocation.

##### A. Deep Q-Network (DQN)

1) *Deep Q-Network background:* In Deep Q-network method, there are mainly two processes, which are the training and the testing. The training process is generated by the interactions with an environment/simulator and the agent in order to observe the agent's state and learn the weights to play the actions that give the best reward. In the testing process, for each time step, the agent observes the environment/simulator, and then accordingly chooses the action based to its trained DQN.

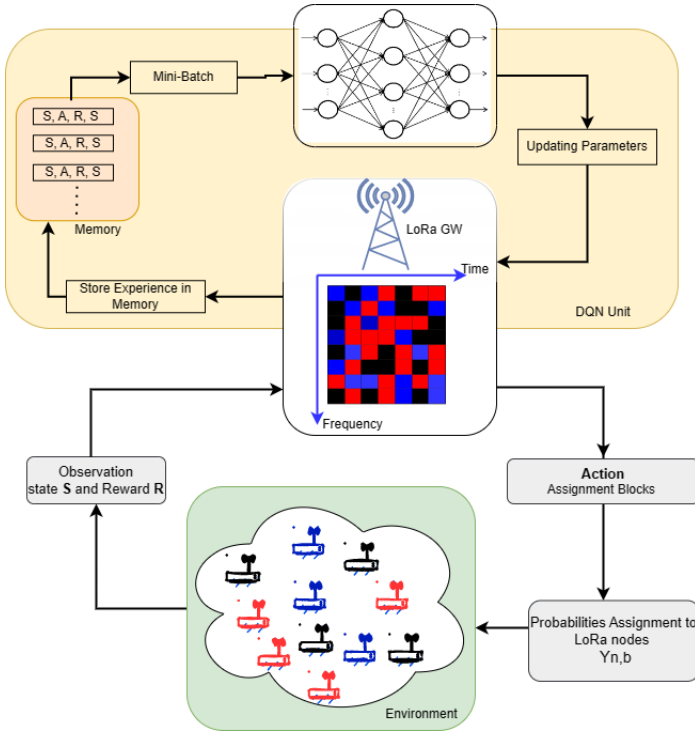


Fig. 1: H-DQN solution

As presented in Algorithm 1, the training starts at first by generating the network parameters, including the LoRa nodes, their parameters, and to which service they belong. The second step consists in initializing the DQN agent, the DQN hyper-parameters. This includes the main DQN and the target DQN and the replay memory  $M$ . The training process consists of several episodes, each of which covers a number of time steps.

In each episode, the environment is built by a random allocation of RBs to the services. In each time step, the agent takes an action for its current state randomly or using the action with the largest Q-value. Afterwards, it updates the allocation of resource blocks to the services. After that, the heuristic algorithm is executed to allocate the probability of the LoRa nodes belonging to a service for each block as described in section (V-B). Next, the environment/simulator provides the reward  $R(t)$  and the next state  $S(t+1)$ .

For each time step, where the DQN makes a decision, the agent's experience  $e_t$ , which is defined as a tuple and  $e_t = (S(t), a(t), R(t), S(t+1))$  and includes state transitions, actions and rewards, is stored in the replay memory  $M$ . As a result, the agent trains the network by sampling a random batch of experiences from  $M$  at each iteration in order to achieve an efficient learning.

Eventually, to calculate the loss function depicted in eq. 9. The Deep Neural Network (DNN) approximates the target Q-values, then the network weights are updated using the gradient descent to reduce the loss function.

$$L(w) = \mathbb{E} \left\{ (y_{DQN}(t) - Q(s, a, w))^2 \right\} \quad (9)$$

where  $y_{DQN}$  denotes the target Q-value and it is given as follows:

$$y_{DQN}(t) = r(s, a) + \gamma \max_{a' \in A} Q(s', a', w) \quad (10)$$

where  $\gamma$  is a discount factor to provide less weight to the future reward,  $r(s, a)$  is the corresponding reward to the state-action pair  $(s, a)$ .

---

**Algorithm 1** DQN Algorithm.

---

**Input:** Agent Gateway, Environment simulator.

**Output:** Trained DQN

- 1: The generation of the network parameters.
  - 2: The initialization of the agent's DQN hyper-parameters.
  - 3: **for** each episode **do**
  - 4:   Initialise:
  - 5:   Random allocation of resource blocks to services
  - 6:
  - 7:   **for** each time step **do**
  - 8:     Choose action  $a_t$
  - 9:      $a_t = \begin{cases} \text{a random action} & \text{with probability } \epsilon \\ \max_a Q^*(s_t, a; w) & \text{otherwise} \end{cases}$
  - 10:     Execute action
  - 11:     Apply the heuristic algorithm, Section (V-B)
  - 12:     The agent Observe Reward  $R_t$  and the next state  $S_{t+1}$ ,
  - 13:     Store the experience  $e_t$  in replay memory  $M$ ,
  - 14:     **if** batch size **then**
  - 15:       samples a random batch of experiences from  $M$ ,
  - 16:       Calculate an estimated Q-value  $Q(s, a)$
  - 17:       Calculate loss function  $L(w)$  Eq.9
  - 18:       Performs a gradient descent to update  $w$ .
  - 19:     **end if**
  - 20:   **end for**
  - 21: **end for**
- 

2) *Resource blocks allocation model:* Based on a scenario containing a single base station with three types of services sharing the RBs, LoRa gateway is considered as the agent that manages the distribution of RBs to services. In the general case, the LoRa Network Server (LNS) will be in charge of such a function.

We define the state space, the available joint actions, the reward function as follows:

**State:** The LoRa gateway agent learns the environment's state at each time step  $t$ . The state at time  $t$  is defined as:  $S(t) = \sum_{j=1}^J S_j(t)$ , whereby  $S_j(t)$  is the state of service  $j$  at time step  $t$ , which is related to the satisfaction of the PDR, and the percentage of RBs used by each service, and it is expressed as:

$$S_j(t) = [\alpha_j^{t-1}, \beta_j^{t-1}] \quad (11)$$

, whereby  $\alpha_j^{t-1}$  refers to the satisfaction of the PDR of service  $j$  at the previous time step  $t$ , and  $\beta_j^{t-1}$  refers to the percentage

of resources occupied by service  $j$  at the previous time step  $t$ .

**Actions:** The set of all possible actions that can be taken by the LoRa gateway agent denoted as:  $A(t) = (a_1^t, \dots, a_k^t)$ . At each time step  $t$ , based on the currently observed state  $S(t)$ , the LoRa gateway agent takes the appropriate action  $a_k^t \in A(t)$  that maximizes the reward  $R(t)$ , and then the environment provides the next state  $S(t+1)$ . Each action includes a percentage of 5%<sup>1</sup> of RBs that must be transferred from one service to another. The set of available actions are:

$$\forall s_1, s_2 \in \mathbb{S}, A(t) = ([s_1 \rightarrow s_2], [s_2 \rightarrow s_1], \{\text{No action taken}\}) \quad (12)$$

The action  $a_1^t = [s_1 \rightarrow s_2]$  consists of an exchange of RBs by transferring 5% of RBs from service  $s_1$  to another service  $s_2$ , while the action  $a_k^t = \{\text{No action taken}\}$  is used when no action need to be taken (i.e., convergence of the algorithm).

**Reward:** After allocating RBs between services by selecting the best action, and allocating the probability of access of LoRa nodes by a heuristic algorithm as described in the next section. The environment returns a reward of the system, which is defined, to guide the gateway agent to make decisions to achieve its goal. The goal is to maximize the PDR of each service while ensuring the priority between the services, which can be expressed as follows:

$$R(t) = \max_{s \in \mathbb{S}} \left( \sum PDR_s \times w_s \right) \quad (13)$$

### B. Proposed heuristic

---

**Algorithm** Heuristic Algorithm.

---

**Input:** Allocated Resource Blocks based on DQN.

**Output:** Allocated probability of access to LoRa nodes.

```

1: function PROBABILITY( $N$ )
2:    $length = len(S_i)$ 
3:    $p = []$ 
4:   for each LoRa node  $n \in S_i$  to  $N$  do
5:      $p[n] = 1 / length$ 
6:   end for
7:   return  $p[n]$ 
8: end function

9: for each time step  $t$  do
10:  for each block  $b_{ij}$  do
11:    PROBABILITY( $N$ )
12:  end for
13: end for

```

---

In this section we propose a heuristic to allocate the access probability of the LoRa nodes belonging to the different services.

At each time step  $t$ , when RBs are allocated among services by choosing the action adopting the DQN algorithm, the

<sup>1</sup>A smaller percentage could be considered, but with the risk of a slower convergence.

heuristic algorithm is designed to allocate the access probability  $p[n]$  of each LoRa node belonging to a service  $S_i$  in each block  $b_{ij}$  allocated for a service with the respect of the extremity of the probabilities as presented in constraints (8d), (8e) and the equi-probability between LoRa nodes of a service as expressed in constraint (8g). After assigning the probabilities to the LoRa nodes, the environment/simulator returns a reward to be used by the DQN, so the system is in the form of a closed loop between DQN and the heuristic to satisfy the needs of each service. The proposed heuristic is presented in algorithm .

## VI. NUMERICAL RESULTS

In this section, we study the performance of our proposed method for resource allocation. The DQN was created and trained in Python 3.9 using the PyTorch framework. The DQN have two fully connected hidden layers with 256 neurons each, as well as an input and an output layers. We used the Rectified linear unit (ReLU) as an activation function, and the Adam optimized with a learning rate  $\beta$ . we consider 300 episodes, each one is composed of 100 time steps, where the resource blocks (RBs) are reset in each episode.

For the simulation, the model built on LoRaSim [13], the LoRaWAN Slicing scenario consists of multiple services sharing the resources blocks of a LoRa gateway GW that is deployed in a square area of  $10 \times 10 \text{ km}^2$ . The LoRa nodes used are randomly assigned to services, with each LoRa node assumed to belong to one and only one service. In our architecture, we consider  $T = 12$  with a 1H slot in the time domain. We also considered three channels for the frequency domain.

RBs are assigned to services, with each block assumed to be assigned to a single service in which the LoRa nodes belonging to it send their packets.

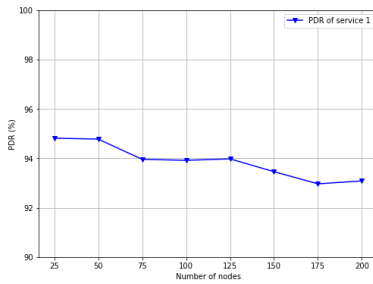
Table I summarizes the different parameters used in the LoRaWAN network.

TABLE I: Simulation Parameters

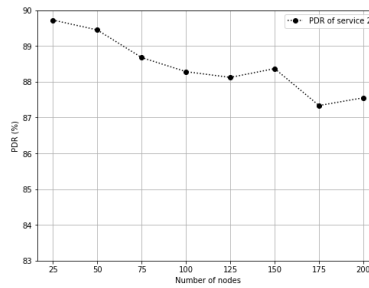
Parameters name	Value
Simulation time	3600000 ms
Duration of time slot (T)	300000 ms
Number of LoRa Gateway	1
Carrier Frequency (CF) / Channels number	868 MHz / 3
Transmission Power (TP)	14 dBm
Spreading Factor (SF)	{7,8,9,10,11,12}
Bandwidth (BW)	125 KHz
Coding Rate (CR)	4/5
Learning rate $\beta$	0.0001
Discount factor $\gamma$	0.99
Mini batch size	64

### A. Loss function Evaluation

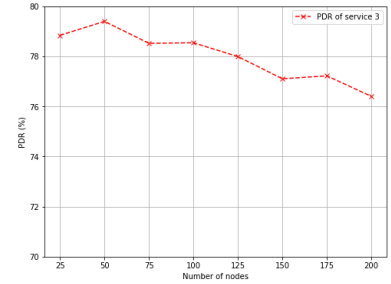
In Fig.3, we show the convergence of the loss function used to train the DQN. The loss function, as shown in Eq. 9, is formed as an indicator that the model is successfully trained. We can see that the values of the loss function reach convergence by decreasing to a small number tending to zero, which proves that the learning method improves with the



(a) PDR of service 1



(b) PDR of service 2



(c) PDR of service 3

Fig. 2: Packet Delivery Rate Evaluation

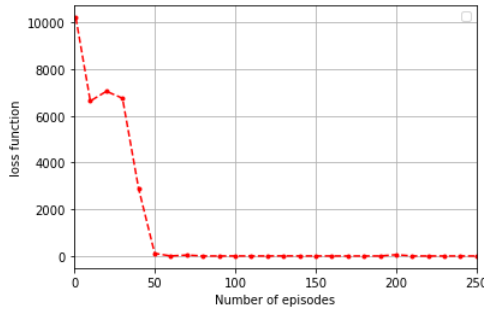


Fig. 3: Loss function evaluation

repetitions of the episodes. This indicates that the optimizer used provides good predictions.

### B. Packet Delivery Rate Evaluation

We notice, in Fig. 2, that the PDR of each service, decreases when the number of nodes increases. This is due to the fact that the number of transmitted packets increases, which puts pressure on the resource blocks. So that in the case of a low number of nodes, there is no huge pressure on the resources assigned to a service, which leads to a better PDR, and therefore many packets will be transferred successfully.

The results in Figures 2a, 2b, and 2c indicate that the first service performs better than the second service and the second service performs better than the third service, this is due to the weight parameter that represents the priorities between services. This means that RBs are assigned to services based on maximizing the PDR while respecting the priority between services.

## VII. CONCLUSION

In this paper, we propose an allocation scheme for slicing the LoRaWAN network using the H-DQN algorithm to guarantee the priority between services and improve the network PDR. First, an inter-service method was proposed using the DQN algorithm to allocate resource blocks to services. Then, an intra-service method was proposed using the Heuristic algorithm to allocate access probabilities to the devices they own within the services. Simulation results showed that the proposed approach improves the Packet Delivery Rate (PDR) while guaranteeing prioritization between services.

## ACKNOWLEDGEMENTS

This work was partially supported by the Intelligentsia, ANR project.

## REFERENCES

- [1] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.
- [2] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of lorawan for iot: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, 2018.
- [3] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale lorawan networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, 2017.
- [4] T. T. Nguyen, H. H. Nguyen, R. Barton, and P. Grossetete, "Efficient design of chirp spread spectrum modulation for low-power wide-area networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9503–9515, 2019.
- [5] L. Alliance, "Lorawan r1. 0 open standard released for the iot," 2015.
- [6] A. S. D. Alfoudi, S. S. Newaz, A. Otebolaku, G. M. Lee, and R. Pereira, "An efficient resource management mechanism for network slicing in a lte network," *IEEE Access*, vol. 7, pp. 89441–89457, 2019.
- [7] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmcc: A communication-theoretic view," *Ieee Access*, vol. 6, pp. 55765–55779, 2018.
- [8] S. Messaoud, A. Bradai, and E. Moulay, "Online gmm clustering and mini-batch gradient descent based optimization for industrial iot 4.0," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1427–1435, 2019.
- [9] S. Dawaliby, A. Bradai, and Y. Pousset, "Adaptive dynamic network slicing in lora networks," *Future generation computer systems*, vol. 98, pp. 697–707, 2019.
- [10] F. Z. Mardi, M. Bagaa, Y. Hadjadj-Aoul, and N. Benamar, "An efficient allocation system for centralized network slicing in lorawan," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2022, pp. 806–811.
- [11] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, "Deep federated q-learning-based network slicing for industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5572–5582, 2020.
- [12] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2016, pp. 59–67.
- [13] "Lorasim." [Online]. Available: <https://www.lancaster.ac.uk/scs/sites/lora/lorasim.html>