Dynamic Parameter Allocation with Reinforcement Learning for LoRaWAN

Mi CHEN, *Student Member, IEEE,* Lynda Mokdad, *Senior Member, IEEE,* Jalel Ben-Othman, *Senior Member, IEEE,* Jean-Michel Fourneau

***Abstract*—LoRaWAN attracted lots of attention with its capacity for large device numbers, long-range and low power consumption. In order to simplify the transmission procedure, a pure Aloha protocol is implemented into its MAC layer. However, as the number of connected devices to the base station increases, the devices' transmission parameters allocation becomes a vital issue related to network performance. This research contributes to the decentralized dynamic Spreading Factor (SF) allocation strategies during transmission by proposing STEPS, a Score Table based Evaluation and Parameters Surfing approach. STEPS is a reinforcement learning-based method that evaluates and changes the parameters based on probability and score tables. It provides a nondeterministic parameter selection method by updating the table while transmitting. Some variants of STEPS with different algorithms are proposed. Moreover, an estimation-based initialization is proposed to improve learning performance. Simulations and statistical tests are carried out with MULANE, a lightweight LoRaWAN Simulator developed in our previous work. The results show that the estimation has a high confidence level. Compared with the baseline methods, the proposed methods reduce energy consumption by 24-27% in different numbers of nodes. For bi-directional transmission, the proposed methods increase the 18% network throughput in a small number of nodes and 33% in a large number of nodes. Moreover, the proposed methods provide a framework of decentralized parameter allocation, which gives the extendability of this work.**

***Index Terms*—LoRaWAN, reinforcement learning, Decentralized Spreading Factor allocation, energy consumption**

I. INTRODUCTION

LoRa is a wireless communication technology that aims to provide end-to-end, energy-efficient communications for the Internet of Things (IoT) and Machine To Machine (M2M) applications [1]. Using a spread spectrum technique with LoRa chips, end devices (EDs) can communicate in a long-range area with a star-of-star topology. Furthermore, operating on the unlicensed ISM band (e.g., 868 MHz in Europe) [2] also makes LoRa a low-cost, easy-implement technology. Thus, LoRaWAN is now widely used in long-range IoT applications such as railway infrastructure monitoring [3] and smart-city monitoring [4]. Moreover, the IoT scenarios with LoRaWAN also show a great potential to be an important application in the future SAGIN (Space-Air-Ground Integrated Network) technologies [5].

In order to reduce energy consumption, most LoRaWAN devices are Class-A devices [6]. The devices spend most of their time in sleep mode and wake up only when they have an uplink packet to send. The uplink packet is sent with Aloha protocol following local duty-cycle restrictions. Once the uplink message is sent, the device opens two windows to receive downlink messages. Depending on the windows' availability, the gateway tries to send an ACK message during one of the two windows. If the device does not receive an ACK message during both windows, it tries to do the retransmission until the ACK message is received or the limit of retransmission times is exceeded [1].

Usually, the node unable to receive the ACK message is caused by the following two reasons:

1) The base station does not receive the packet due to collision, packet loss, or other reasons.
2) The base station received the packet, but due to the duty-cycle limitation, it was unable to find a suitable radio resource to send back an ACK message

Since LoRaWAN has a MAC protocol based on the Aloha mechanism, it faces a great challenge of the packet loss rate and collision [7], which leads to transmission failure caused by reason 1.

In order to network performance, LoRaWAN has adopted some mechanisms to avoid collision and packet loss. For example, LoRaWAN uses near-orthogonal transmission parameters to reduce interference and collisions, such as the Spreading Factor (SF) [8]. In LoRaWAN transmission, a lower SF often can get a higher data rate in a shorter range. On the contrary, a higher SF can reduce the probability of lost packets but leads to a lower transmission speed and waste radio resources [9]. Furthermore, if too many devices choose high SFs in transmission, the channels are occupied for a long time, causing transmission failure in reason 2. Thus, parameter allocation becomes a vital issue of the performance in LoRaWAN.

Motivated by improving the performance of LoRaWAN, this study focuses on the decentralized dynamic parameters allocation (SF) algorithm to increase the network capability. Some dynamic approaches are introduced in this study in

This research is supported by ASPIRE, the technology program management pillar of Abu Dhabis Advanced Technology Research Council (ATRC), via the ASPIRE Visiting International Professorship program. Reference number ASPIRE/ZU R22036 EU2105.

Mi CHEN and Lynda Mokdad are with Univ Paris Est Creteil, LACL, F-94010 Creteil, France (e-mail: {mi.chen, lynda.mokdad}@u-pec.fr).

Jalel Ben-Othman is with Univ Paris-Saclay, CNRS, CentraleSupélec Lab. L2S, 91190, Gif-sur-Yvette, France, and is also with Univ Sorbonne Paris Nord, and is also with College of Technological Innovation, Zayed University, UAE (e-mail: jalel.benothman@l2s.centralesupelec.fr).

Jean-Michel Fourneau is with Univ. Paris Saclay, UVSQ, lab. DAVID, France (e-mail: jean-michel.fourneau@uvsq.fr).

response to this issue. These approaches allow the devices to dynamically choose their transmission parameters (SF, for example) during the transmission. The main contributions of this study are as follows:

- Firstly, two random dynamic approaches are proposed to avoid choosing bad SF for transmission.
- Then, by integrating a score table and the reinforcement learning idea, STEPS is investigated. It is based on a score table evaluation and parameters surfing approach of LoRaWAN. The devices are considered intelligent agents that follow a transmission, evaluation and update, and surfing approach for a complete transmission. Compared with the traditional method, it provides a nondeterministic selection method. More importantly, STEPS provides a framework of decentralized methods. This brings the extendability of our work. Based on the framework, several extension works are carried out as follows.
- By estimating the success probability of transmission, a decentralized initialization method is proposed and implemented to STEPS to accelerate the learning process.
- By investigating and implementing different reinforcement models and algorithms, several variants of STEPS are also proposed in this study. By modeling the surfing phase as EE (Exploration-Exploitation) problem, STEPS with ϵ-Greedy Algorithm and STEPS with Boltzmann Exploration Algorithm are proposed. By modeling the evaluation and update and surfing phases as an MDP (Markovian Decision Process), STEPS with Q-learning Algorithm and STEPS with Q-learning and policy-based Hybrid Algorithm are proposed.
- Simulation and statistical tests are carried out by simulator MULANE, a LoRaWAN simulator developed in our previous work. The results show that the new methods increase the capability of the LoRaWAN network in energy consumption compared with baseline methods. Moreover, the proposed STEPS methods outperform in other aspects, such as the packet success rate, the packet delivery rate, and the network throughput.

The rest of this paper is organized as follows. Section II presents the related works of parameter allocation in LoRaWAN. In Section III, we present the methods and propose the variants of STEPS. The statistical tests, simulations, and performance evaluation of the proposed methods are given in Section IV. The main contribution of this study and perspectives are given in Section V.

II. RELATED WORKS

The methods for SF allocation can be divided into two types. One is the one-time allocation. In these methods, the devices allocate their SF before the transmission begins. In [10], the author used an algorithm based on the distance between the devices and the base station. In [11], an allocation method based on K-means is proposed. Using the K-means clustering algorithm, the network coverage area is unevenly divided into six parts corresponding to six SFs. Simulations prove that this method is more effective than uniform division. However, this division is centralized and requires a network controller to perform the clustering algorithm and inform the nodes of the clustering center or clustering results, thus increasing communication costs. Therefore, decentralized and distributed allocation methods are needed. In [12], [13] the authors proposed their decentralized methods of initial SF allocation. The nodes choose their initial SF based on either minimum energy consumption [13] or minimum data collection time [12]. In [14], the authors consider scenarios in multi-gateways and propose a resources allocation scheme based on the multilayer virtual cell concept, and the simulation results proved the improvements to the network.

The one-time methods have their bottlenecks. With the lack of channel or environmental information, the one-time methods are sometimes useless and cause more collisions. Moreover, when the allocation is not optimal and causes a collision and loss, the network sticks to this allocation and significantly reduce transmission efficiency. Facing this issue, the authors in [12], [15] proposed new transmission scheduling. However, this makes the algorithms hard to be implemented and incompatible with the classical Aloha method since the proposed methods do not follow the Aloha protocol in LoRaWAN.

Another type of SF allocation is dynamic allocation. The methods in this type allow the node to change the SF during the transmission. Semtech Corporation, the developer of LoRa proposed ADR (Adaptive Data Rate) method in [16], [17]. It is a semi-centralized method. The network server can allocate the nodes' transmission parameters by sending a downlink MAC command based on the received signal-to-noise-ratio (SNR). Moreover, the nodes can also adjust the parameters if the ACK is not received. In [16], the ADR algorithm at the server side is proposed, and the ED-side algorithm is proposed in [17]. Firstly, the server uses the maximum SNR of the last 20 records to allocate the parameters. Considering that the transmission frequency in LoRaWAN is often several minutes, this algorithm takes a long learning process. Secondly, the parameter changing policy is deterministic and memoryless. The ED-side adjustment can only decrease the SF by 1 when it decides to change, while the server-side allocation can only increase the data rate by 1.

In order to improve the performance of ADR, some variants and new schemes are proposed. In [18], the authors proposed Fair-ADR (FADR). It allocates the SF based on the fair collision probability. The authors in [19] proposed BE-LoRa, a server-side approach to adjust the parameters using a non-cooperative game model based on energy consumption. In [20], the authors proposed a semi-centralized method named R-ARM based on the threshold of the SNR. However, these methods are all centralized or semi-centralized and deterministic. Moreover, most methods are memoryless or have limited memory on the ED side. When the downlink message is not received, the nodes risk being stuck in an inappropriate parameter or parameter selection policy.

In order to allocate the parameters in a decentralized way, the developers of LoRaWAN modified the ADR algorithm and proposed Blind-ADR (BADR) in [21]. In this method, nodes take turns using three SFs for transmission. However, this is still a deterministic and memoryless method. Nodes

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

cannot dynamically adjust the SF selection strategy based on previous transmission experience. Moreover, this method only uses three SFs of 7, 10, and 12, increasing the network's collision probability.

The idea of reinforcement learning is used to better make the dynamic decision of parameter selection strategy in a decentralized way. The Q-learning algorithm is implemented in the nodes by the author of [22]. However, the author used this algorithm for the scheduling and the transmission probability rather than the parameter allocation. For the parameters allocation, the authors in [23]–[25] modeled a Multi-Armed Bandit (MAB) problem and implemented the resolving algorithm to the node. In [23], the problem aims to channel allocation and assumes that all devices use the same SF. In [24], the authors ignored the capture effect and assumed a uniform device distribution to model the allocation problem. The author of [25] proposed an SF changing strategy using the EXP3.S algorithm to solve the MAB problem. However, the lack of punishment in these methods makes the devices stay in the same strategy without improvement when failure transmission occurs and reduce the network performance. Furthermore, these methods did not consider the initial allocation problem or strategy design. Thus, a learning process is inevitable for these methods.

In our previous work [26], STEPS - a Score Table-based Evaluation and Parameters surfing approach of LoRaWAN is proposed. It is a decentralized dynamic allocation method with a score table to exploit transmission experience when making a decision. The nodes make a transmission, evaluation and update, and surfing approach based on the previous experiences given by the score table. STEPS also implemented a punishment mechanism in its algorithm, making learning more reliable. Simulation results show that STEPS brings the LoRaWAN network a remarkable improvement for bi-directional transmission by increasing 10-15% packet success rate.

However, there is still room for improvement in the original STEPS. In the next section, some decentralized dynamic methods are investigated. By integrating more reinforcement learning algorithms, three variants of STEPS are proposed as the main contribution of this paper.

III. PROBLEM DESCRIPTION AND TRANSMISSION MODEL

By considering the nodes in the network as intelligent agents, some dynamic behaviors are proposed in this study to improve the network performance. Since the agents choose the strategy during the transmission, the nodes must know whether or not their packets are delivered successfully. In order to get transmission status, the considered problem in this study is based on bi-directional transmission, which means a downlink ACK message is required so that the nodes can make their decision.

The model of the nodes' transmission is shown in Fig. 1. Nodes send an uplink message with Aloha protocol following duty-cycle restrictions. When uplink transmission finishes, nodes wait one second ($RECEIVE\_DELAY1$) and open a short receive window (Rx1) to receive the ACK message. If the ACK message is not received during the first window, nodes open another downlink window (RX2) two seconds after the end of the UL transmission noted $RECEIVE\_DELAY2$. If no downlink message arrives during either of the two windows, nodes try to retransmit until the acknowledgment message is received or the limit of retransmission times is exceeded [1]. For each transmission (including new packet and retransmission attempts), nodes can change the SF with specific policies.

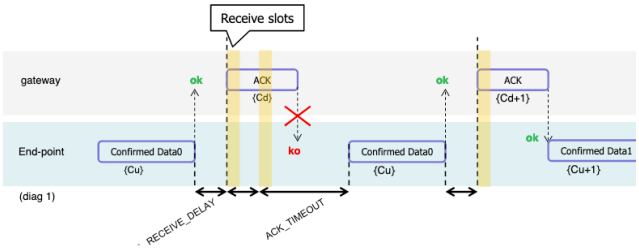


Fig. 1: Transmission model of nodes [1]

Based on the model above, several methods are proposed in the following section.

IV. PROPOSED METHODS

A. Random surfing

When a node detects transmission failure, which means the node does not receive the ACK message in both two downlink windows, the first straightforward idea of the agent is to change the strategy of current parameters to avoid further failure. Based on this idea, the method of random surfing is proposed. The random surfing method is simple and can easily be implemented in devices. When transmission failure occurs, the node considers the current SF choice inappropriate and change the strategy. However, it does not know any information on other parameters. In this case, the node randomly chooses another SF to avoid failure.

However, there are some shortcomings in changing the strategy with every failure. For example, when the failure is caused by reason 2 mentioned in Section I, the limitation of the network capability becomes a great cause. It can be even worse if a node changes to a higher SF since the ACK is still lost, and the energy consumed increases.

B. p-random surfing

In order to avoid changing SF too frequently, the p-random surfing method is proposed. The nodes with this method take the same transmission procedure as the class-A devices. When a transmission failure is detected, the nodes change the chosen SF in a probability p. Once a node decides to change its strategy of SF choice, it chooses randomly from SF7 to SF12 except the current one.

Fig. 2 shows the transmission procedures of the two methods mentioned above. Note that when the threshold $p = 0$, the p-random surfing is equivalent to a static method where the SF choice is fixed, and when p is set to 1, the p-random surfing becomes the random surfing method.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

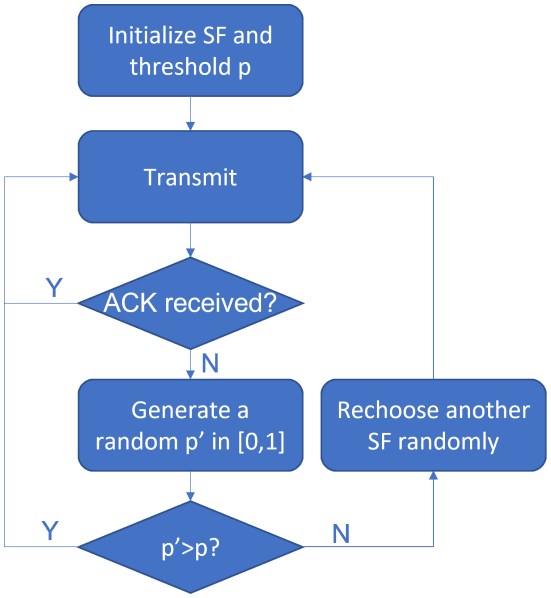


Fig. 2: Transmission approach of random surfing and p-random surfing

However, both of the methods mentioned above still choose parameters randomly. It can be worse if the device changes its SF in some cases. When a device changes to another SF, an SF too high causes a waste of resources, and an SF too low makes the packet lost again in the subsequent transmission. Thus, it is necessary to evaluate the SFs to make a better strategy.

Based on the reinforcement learning concepts, the STEPS approach is proposed. This method is based on a score table that aims to provide more transmission experience for the nodes to decide. Instead of a traditional approach, nodes perform a transmission, evaluation and update, and surfing approach based on the previous experiences given by the score table. The transmission in STEPS still follows the Aloha protocol of LoRaWAN, which makes STEPS easy to be implemented.

C. Original STEPS

The original STEPS is proposed in [26]. It is the first version of STEPS. Fig. 3 shows a structure overview of the method. When joining the network, nodes choose the initial SF using the method mentioned in [10]. When the SF is selected, nodes use the K-nearest neighbor score table method to initialize the score table if nodes have a pre-implemented database. Otherwise, nodes set up an effortless score table based on the initial SF. For example, if the initial SF is 7 or 8, then the score table is set as the following table:

SF7	SF8	SF9	SF10	SF11	SF12
0.498	0.498	0.001	0.001	0.001	0.001

After initialization, nodes start the STEPS approach. The approach includes the transmission phase, the evaluation and update phase, and the parameter surfing phase.

In the transmission phase, the STEPS nodes follow the same Aloha protocol as the regular LoRaWAN network. Based on the ACK of the transmission, nodes evaluate the transmission and update the score table in the evaluation and update phase. The original STEPS uses three coefficients of updating where:

- c_a is the ACK reward that the node takes when it receives an ACK.
- c_r is the only-receive punishment representing the punishment of reason 2 mentioned in Section I.
- c_f is the fail punishment that is taken by the node when the transmission fails.

When the node does not receive the ACK message, it first estimates that the loss is caused by which reason mentioned above. Then, based on the estimated probability, the node chooses the corresponding coefficient and updates the score table.

After updating the score table, nodes go to the surfing phase. In the original STEPS, the score table is normalized to 1 to represent the probabilities of choosing each SF. Thus, nodes decide based on the score table in the surfing phase. Moreover, the output of the evaluation and update phase is used in the surfing phase for a "further punishment" mechanism to avoid a long punishment process when a previous strategy is no longer applicable to the current network for sudden reasons. In addition, STEPS also defines a mechanism to set up the downlink SF to make full use of the downlink radio resource.

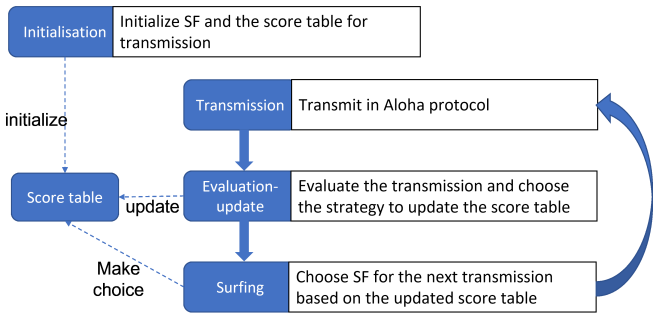


Fig. 3: Structure of STEPS

With the above phases, nodes can choose the SF for each transmission based on the score table. The transmission feedback can also update the score table to let the nodes make decisions more appropriately. The independence of the nodes makes STEPS a decentralized method. These features make STEPS compatible with traditional LoRaWAN equipment and easy to deploy into the LoRaWAN network.

In [26], the simulation results show that the original STEPS remarkably improves the LoRaWAN network in ACK ratio for bi-directional transmission.

However, the original STEPS has some shortcomings. Since selecting a higher SF can get a higher transmission success rate, a higher SF in the score table accumulates a higher score under the same reward coefficient. This makes the node tend to choose a higher SF as the transmission strategy while ignoring the problem of high energy consumption caused by the high SF.

In addition, the selection of the initial SF in the original STEPS is only based on the distance from the node to the base station, and the establishment of the initial score table is

too simple. Therefore, the node may choose a poor strategy and build a poor score table, wasting transmission resources and affecting network performance.

Facing these problems, we proposed the following variants of STEPS by including some other reinforcement learning algorithms.

D. Initialization of STEPS with Prior-knowledge

In [27], we proposed an initialization method based on an MDP (Markovian Decision Process) model. With the optimal path obtained by solving the model, nodes can choose their initial SF and set up the initial score table. Simulation results showed that the nodes spend less energy consumption with this initialization than the original STEPS. However, this method needs the network's global information, such as nodes' distribution. Due to limitations in computing performance and power consumption, sometimes nodes cannot predict general network information and the overall strategies of others.

In order to have a more general initialization for the nodes, an initialization method with prior knowledge is proposed. In this method, the nodes are allowed to know some of the prior environmental knowledge, such as the path loss model parameters. With this knowledge, nodes can decide the SF choosing strategy and establish the initial score tables. The node first estimates the success probability based on the knowledge. Then, it chooses the initial SF as little as possible while keeping the probability higher than a threshold. The node can establish the score table with an exponential decay with the initial SF chosen.

Unlike the assumptions in [27], nodes in this method cannot predict whether the gateway can send ACK messages correctly. Moreover, nodes can neither know whether the transmission is collided without others' information. Therefore, the node can only assume that all uplink packets without loss can be answered. In this case, the probability without a loss is used as an indicator of the success probability.

In LoRaWAN transmission, an uplink message is not lost only when the received signal-to-noise ratio (SNR) is greater than the specific threshold q_{SF}. Thus, the probability $H_i(SF)$ that the packet sent by node i at a distance d_i with the spreading factor SF is not lost can be expressed as Equation (1) [28].

$$H_i(SF) = \mathbb{P}[SNR \geq q_{SF} | d_i] \tag{1}$$

Authors in [11], [28], [29] give the theoretical calculation of Equation (1). Assuming that the channel gain is modeled as an exponential random variable with an average of 1, which means $h_i \sim exp(1)$, the probability $H_i(SF)$ that the uplink packet sent by node i is not lost can be expressed as:

$$H_i(SF, d_i) = exp\left(-\frac{\mathcal{N} q_{SF}}{P_i g(d_i)}\right) \tag{2}$$

In Equation (2), $\mathcal{N} = -174 + NF + 10\log_{10}(Bandwidth)$ dBm, NF is the receiver noise which is often 6 dB, P_i is the transmission power of node i, and $g(d_i) = \frac{\lambda^2}{(4\pi d_i)^\eta}$ is the path loss attenuation function, where λ is the wavelength and η is the path loss exponent typically taken at 2.7 in urban environments or 4 in suburban environments [29].

In this work, STEPS with Prior Knowledge Initialization uses Log-distance path loss model for the SNR calculation. This model is also used in [30]–[32], [32], [33]. In this model, the SNR is modeled in dBm with Equation (3) where P_{tx} is the transmit power in dBm, PL_0 is the distance path loss d_0, η is the loss exponent and X_σ is the attenuation (in decibels) caused by flat fading which is modeled with a Gaussian random variable $X_\sigma \sim N(\mu, \sigma^2)$ with $\mu = 0$.

$$\begin{aligned} SNR &= P_{rx} - \mathcal{N} \\ &= P_{tx} - \left(PL_0 + 10\eta \log_{10}\left(\frac{d_i}{d_0}\right) + X_\sigma\right) - \mathcal{N} \end{aligned} \tag{3}$$

With Equation (3), the probability of H_i can be written as follows:

$$\begin{aligned} &H_i(SF, d_i) = \mathbb{P}[SNR \geq q_{SF} | d_i] \\ &= \mathbb{P}[X_\sigma \geq q_{SF} + \mathcal{N} - P_{tx} + PL_0 + 10\eta \log_{10}\left(\frac{d_i}{d_0}\right) | d_i] \end{aligned} \tag{4}$$

In [27], the calculation of $H_i(SF, d_i)$ is given. By noting the Cumulative Distribution Function(CDF) of normal distribution $F_{X_\sigma}(x) = \frac{1}{2}(1 + erf(\frac{x}{\sigma\sqrt{2}}))$, the probability of H_i can finally be as Equation (5):

$$\begin{aligned} &H_i(SF, d_i) \\ &= \frac{1}{2}\left(1 - erf\left(\frac{q_{SF} + \mathcal{N} - P_{tx} + PL_0 + 10\eta \log_{10}\left(\frac{d_i}{d_0}\right)}{\sigma\sqrt{2}}\right)\right) \end{aligned} \tag{5}$$

Statistic tests results show the credibility of the estimation of prior knowledge. Therefore, the initialization steps of this STEPS variant is as shown in Algorithm 1.

Algorithm 1 Initial SF choosing with prior knowledge

Input: d_i: distance of node i from base station,
$q_{SF}, \mathcal{N}, P_{tx}, PL_0, \eta, \sigma, d_0$: measured environmental parameters and fixed node self-parameters
H_{thres}: threshold of minimum SF
Output: SF_{init}: The initial SF chosen

```
Set SF_init = 12
for sf ∈ ⟦7, 12⟧ do
    Calculate H_i(sf, d_i) with Equation (5)
    if H_i(sf, d_i) ≥ H_thres then
        Set SF_init = sf
        return SF_init
    end if
end for
return SF_init
```

Next, in the score table initialization part, the SF_{init} given by Algorithm 1 is the smallest SF that makes the success probability over the threshold. Therefore, the score of the SFs smaller than SF_{init} can be set directly to 0 to prevent the node from choosing a strategy that causes high loss probability. Furthermore, since higher SF always leads to a longer transmission time and energy consumption, it is unwise to give the same initial score to high SFs as the low SF.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

However, the scores of high SFs should be greater than 0 since they give a high success probability.

Thus, an exponential decay method is used to initialize the score table in STEPS with Prior Knowledge Initialization. In this method, after selecting SF_{init}, the node sets the scores of all SFs lower than SF_{init} to 0. The initial scores of SFs higher than SF_{init} decreases to encourage the nodes to select the lowest SF strategy while ensuring transmission success.

The score table initialization approach is given in Algorithm 2.

Algorithm 2 Initial score table with prior knowledge

Input: SF_{init}: The initial SF chosen,
α: Exponential decay parameter
Output: ST: The initial score table

Set $ST = \{0, 0, 0, 0, 0, 0\}$
for $sf \in [\![7, 12]\!]$ **do**
  if $sf \geq SF_{init}$ **then**
    Set $ST_{sf-7} = e^{-\alpha|SF_{init}-sf|}$
  else
    Set $ST_{sf-7} = 0$
  end if
end for
Set $S = \sum_{s \in ST} s$
for $s \in ST$ **do** $s = \frac{s}{S}$
end for
return ST

Then, unlike the original STEPS, the update coefficients varies with the different SF in the evaluation and update phase. The smaller SF gets a higher reward coefficient when the node successfully receives the ACK message. On the opposite, when the node does not receive the ACK message because of reason 2 mentioned above, the parameter selection is not necessarily the main reason for the transmission failure. The penalty also decreases as the SF decreases. However, when a transmission fails for reason 1, all SFs still get the same penalty since the node parameter selection is inappropriate. These changes make the node always try to select a lower SF while ensuring transmission success. The complete evaluation and update phase is shown as Algorithm 3.

Moreover, considering zero values are in the score table, the "further punishment" mechanism in STEPS with Prior Knowledge Initialization is also modified to avoid the appearance of all-zero score tables. Algorithm 4 shows the surfing phase, and Algorithm 5 shows the complete procedure of the STEPS approach.

Other variants are proposed as extensions of STEPS by investigating several reinforcement learning algorithms. The proposed method are given in the following parts.

E. Variants of STEPS with EE Problem

In the reinforcement learning theory, the surfing phase of STEPS can also be modeled as an EE (Exploration-Exploitation) problem. The EE problem is a problem of whether the intelligent agents choose the best strategy based on existing experience (Exploitation) or explore new strategies to increase future benefits (Exploration) when making decisions. By implementing two existing algorithms of the EE problem into the evaluation and update phase and the surfing phase, two other STEPS variants are proposed as follows.

Algorithm 3 Evaluation and update phase with prior knowledge

Input: n: Current node, ST: Current score table of the node, sf: Current SF of the packet,SF_{init}: The initial SF chosen, c_a,c_r,c_f: The updating coefficients
Output: flag: Whether the node will go into "further punishment" in surfing phase

The transmission is finished
if n received ACK during the downlink windows **then**
  $ST_{sf} = ST_{sf} \times (1 + c_a e^{-|sf-SF_{init}|})$
  Let flag = False
else
  Let $prob = ST_{sf}$
  Let $c = c_r$ *or* c_f with a probability $prob$
  if $c = c_f$ **then**
    $ST_{sf} = ST_{sf} \times c$
    Let flag = False
  else
    $ST_{sf} = ST_{sf} \times [(c_r - c_f)e^{-|sf-SF_{init}|} + c_f]$
    Let flag = True
  end if
end if
$S = \sum_{s \in ST} s$
for $s \in ST$ **do** $s = \frac{s}{S}$
end for
return flag

Algorithm 4 Surfing phase with prior knowledge

Input: sf_{last}: The strategy who won the surfing approach the time before the last, sf: Current SF of the packet, ST: Current score table of the node, flag: the return of last phase, β:the further punishment coefficient

if $sf_{last} = sf$ and $flag = True$ **then**
  $ST_{sf} = ST_{sf} \times \beta$
  Normalize ST with the same method as Algorithm 3
end if
$sf_{last} = sf$
Set the value of sf with the probabilities of ST

1) STEPS with ϵ-Greedy Algorithm:

The first variant of STEPS with EE idea is STEPS with ϵ-Greedy Algorithm. It is a straightforward algorithm. In the evaluation and update phase and the initialization, the node evaluates the transmission and update the score with the same method as the STEPS with Prior Knowledge Initialization. When the node decides SF in the surfing phase for the subsequent transmission, it chooses the optimal strategy in the updated score table with a probability of $1-\epsilon$, and it may also make the decision randomly ignoring the score table with the probability of ϵ.

In addition, a pruning strategy is carried out in the surfing phase to prevent the node from choosing the SF that is highly

Algorithm 5 STEPS with prior knowledge

Input: n: Node initialized by Algorithm 1 and Algorithm 2

1: **while** True **do**
2: n transmits by Aloha in uplink when there is a packet and open the downlink windows
3: n evaluates the transmission after the windows close and update with Algorithm 3
4: n surfs among the parameters and choose a strategy by Algorithm 4
5: **end while**

probable to be lost. When the node decides to explore other strategies, it gives up all the strategies that are less than SF_{init} calculated during the initialization. Thus, the probabilities P_{sf} of the strategies are given by Equation (6), and the complete approach of STEPS with ϵ-Greedy Algorithm is given in Algorithm 6.

$$P_{sf} = \begin{cases} 1-\epsilon+\dfrac{\epsilon}{K}, \ if \ sf = \arg\max\limits_{sf}(ScoreTable_{sf}) \\ \dfrac{\epsilon}{K}, \ else \ if \ sf \geq SF_{init} \end{cases} \\ K = 13 - SF_{init} \tag{6}$$

Algorithm 6 STEPS with ϵ-Greedy Algorithm

Input: n: Node initialized by Algorithm 1 and Algorithm 2

1: **while** True **do**
2: n transmits by Aloha in uplink when there is a packet and open the downlink windows
3: n evaluates the transmission after the windows close and update with Algorithm 3
4: **for** $sf \in [\![SF_{init}, 12]\!]$ **do** n calculates P_{sf} with Equation (6)
5: **end for**
6: n surfs among the parameters and choose a strategy based on the P_{sf} calculated.
7: **end while**

It should be noted that the choice of the parameter ϵ is sensitive. If ϵ is too large, it causes too much randomness in decision-making. For example, when $\epsilon = 1$, the method becomes a pure exploitation algorithm near to random surfing method mentioned before. On the other hand, if ϵ is too small, it leads to insufficient exploration. For example, when $\epsilon = 0$, the node always chooses the SF with the highest score in the score table.

2) STEPS with Boltzmann Exploration Algorithm:

Another variant of STEPS with EE idea is STEPS with Boltzmann Exploration Algorithm. This method uses a temperature parameter τ to control the ratio of exploration and exploitation during the surfing phase.

Equation (7) shows the exploration and exploitation strategy calculation in the Exploration Algorithm.

$$P_{sf} = \frac{exp\left(\frac{ScoreTable_{sf}}{\tau}\right)}{\sum\limits_{s \in ScoreTable} exp\left(\frac{s}{\tau}\right)} sf \in [\![SF_{init}, 12]\!] \tag{7}$$

The SF_{init} and the $ScoreTable$ in Equation (7) is initialized and updated with the same initialization method as STEPS with Prior Knowledge Initialization. This means this approach takes the same initialization method and evaluation and update phase as STEPS with Prior Knowledge Initialization. Thus the algorithm of STEPS with Boltzmann Exploration Algorithm is as shown in Algorithm 7.

Algorithm 7 STEPS with Boltzmann Exploration Algorithm

Input: n: Node initialized by Algorithm 1 and Algorithm 2

1: **while** True **do**
2: n transmits by Aloha in uplink when there is a packet and open the downlink windows
3: n evaluates the transmission after the windows close and update with Algorithm 3
4: **for** $sf \in [\![SF_{init}, 12]\!]$ **do** n calculates P_{sf} with Equation (7)
5: **end for**
6: n surfs among the parameters and choose a strategy based on the P_{sf} calculated.
7: **end while**

Note that when the temperature parameter $\tau \to 0$, the above method is pure exploitation; when $\tau \to \infty$, the above method is pure exploration. Therefore, the value of τ can be set to control the ratio of exploration and exploitation. Some studies also refer to $\frac{1}{\tau}$ as the learning rate.

F. Variants of STEPS with MDP Problem

Another commonly used reinforcement learning model is MDP (Markovian Decision Process). STEPS is modeled in this section as a partially observable MDP (POMDP) model. Moreover, combined with the prior-knowledge in the previous subsection, two variants of STEPS are proposed: STEPS with Q-learning Algorithm and STEPS with Q-learning and policy-based Hybrid Algorithm.

1) MDP model for STEPS:

The Markov Decision Process (MDP) is a mathematical model of sequential decisions [34]. It can be used to simulate the randomness strategies and rewards of the agent in an environment where the system state has Markov properties, as shown in Fig. 4. In STEPS, the nodes are considered as agents. At each step, the agents take action (transmission with a chosen SF) and enter a new state(transmission success or failure). The agents can update their strategies and take a new action in the subsequent transmission with the reward defined.

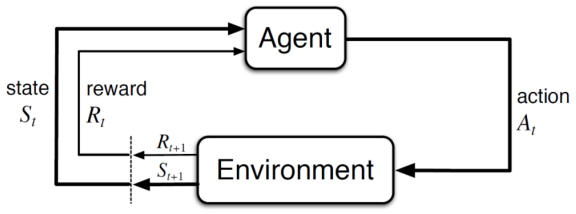


Fig. 4: Interaction of agent and environment [34]

A traditional MDP can be described as a tuple $M =< S, A, P, R >$ where:

- S is the finite set of states. In STEPS, $S = \{Succ, UL-Fail, Only-received\}$. $Succ$ represents the state that the previous transmission is successful, $UL-Fail$ represents that the node did not receive ACK for the previous transmission due to reason 1 mentioned in Section I, and $Only-received$ represents that the node did not receive ACK for the previous transmission, but the uplink transmission succeeded.
- A is the set of actions. With the prior-knowledge proposed above, when the agent initializes its initial SF SF_{init} using Algorithm 1, the set A can be defined as $A = \{SF_{init}, .., SF_{12}\}$ The actions SF_{sf} represent that the node at state $s \in S$ takes sf as its SF for next transmission.
- $P : S \times A \times S$ is the matrix of transitions. The elements $\mathbb{P}(s, a, s') \in P$ are the transition probabilities of states, which represent the probabilities of reaching state $s' \in S$ from state $s \in S$ by the action $a \in A$.
- $R : S \times A \times S$ is the reward matrix. The elements in $r(s, a, s') \in R$ is the expected reward of reaching state $s' \in S$ from the state $s \in S$ with action $a \in A$.

However, in the LoRaWAN network, agents can only know their state with the reception of the ACK message. This means the states $UL-Fail$ and $Only-received$ are unobservable to the agents. Thus, a partially observable MDP (POMDP) is modeled. As shown in Fig. 5, in this MDP, the states in set S are also called trusted states. A finite set of observed states $O = \{Succ, ACK-lost\}$ is added to the tuple M where $Succ$ is the same state as $Succ \in S$, and $ACK-lost$ is the state that the agent did not receive the ACK message for the previous transmission. Every time the agent observes the state $ACK-lost$, it estimates the probability that the uplink is successful with Equation (5) with current SF. Then, the agent goes to one of the states $UL-Fail$ or $Only-received$ with the calculated probability.

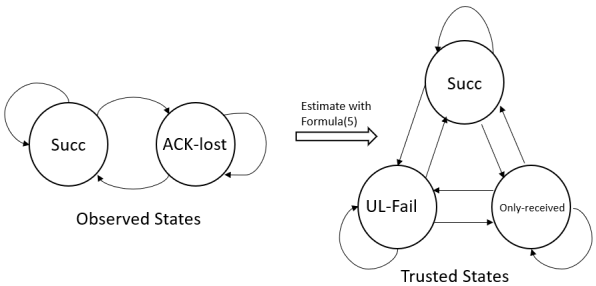


Fig. 5: The partially observed MDP model of STEPS

Moreover, the matrix P in the tuple M is unknown to the nodes and often impossible to be estimated. In this case, model-free algorithms are more commonly used. In the following part, two other STEPS variants are proposed by implementing two model-free algorithms and the MDP modeled above into the evaluation and update phase and the surfing phase.

Algorithm 8 Initial score table of Q-learning

Input: SF_{init}: The initial SF chosen,
Output: ST: The initial score table
1: Set $ST = \{\{0\}^{12}_{SF_{init}}, \{0\}^{12}_{SF_{init}}, \{0\}^{12}_{SF_{init}}\}$
2: **for** $state \in S$ **do**
3: **for** $sf \in [\![SF_{init}, 12]\!]$ **do**
4: Set $ST(state, sf) = 2^{12-sf}$
5: **end for**
6: **end for**
7: **return** ST

2) STEPS with Q-learning Algorithm:

According to the characteristics of the Score table in STEPS, Q-learning is the first to be considered and implanted in the model-free MDP algorithm. Q-learning is a value-based algorithm in the reinforcement learning algorithm. STEPS with Prior-knowledge and Q-learning algorithm is proposed by combining STEPS with the Q-learning algorithm. In this variant, agents construct their score table for all combinations of (s, a) where state s in trusted states S and $a \in A$. The value $Q(s, a)$ in the table represents the expectation of reward if agent takes action a at state s. When the node joins the network, it initializes its SF with Algorithm 1. Then, it constructs the score table (i.e., Q-table) with an exponential decay as shown in Algorithm 8. The example of the initialized table with $SF_{init} = 7$ is as follows:

	SF7	SF8	SF9	SF10	SF11	SF12
Succ	32	16	8	4	2	1
UL-Fail	32	16	8	4	2	1
Only-received	32	16	8	4	2	1

Then, in the evaluation and update phase, nodes first estimates their trusted state s' based on the state observed. With the estimated trusted state s', the current state s, and the action a, nodes update their score table with Equation (8), which is the same as the original Q-learning algorithm [35].

$$Q(s, a) := (1 - \alpha)Q(s, a) + \alpha[r(s, a, s') + \gamma \max_{a' \in A} Q(s', a')] \tag{8}$$

In Equation (8), α is the learning rate, γ is the discount factor. The reward $r(s, a, s')$ is defined as Equation (9). As mentioned before, the idea that different SF with different rewards is also used. The smaller SF with higher reward when $s' = Succ$. The penalty decreases as the SF decreases when $s' = Only-received$. The same penalty is still given to all SFs when $s' = UL-Fail$.

$$r(s, a, s') = \begin{cases} 1 + c_a e^{-|a - SF_{init}|}, if\ s' = Succ \\ c_r e^{-|a - SF_{init}|}, if\ s' = Only-received \\ c_f, if\ s' = UL-Fail \end{cases} \tag{9}$$

Algorithm 9 shows the evaluation and update phase of STEPS with Q-learning algorithm.

In the surfing phase, Q-learning takes a value-based strategy. The node at state s' chooses the SF a^* where $Q(s', a^*) = \max_{a \in A} Q(s', a)$. Furthermore, in Q-learning, the ϵ-Greedy is also implemented to give the node a chance for exploration.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

Algorithm 9 Evaluation and update phase with Q-learning

Input: n: Current node, ST: Current score table of the node, sf: Current SF of the packet,SF_{init}: The initial SF chosen, c_a,c_r,c_f: The updating coefficients, s:current state
Output: s

The transmission is finished, n observe the ACK message for observed state o
if ACK received (i.e., $o = Succ$) **then**
    $s' = Succ$
else
    Calculate $prob$ with Equation (5)
    Let $s' = Only-received$ or $UL-Fail$ with a probability $prob$
end if
Get $r(s, a, s')$ with Equation (9) using s',c_a,c_r,c_f
Update $Q(s, a)$ with Equation (8) using s',a,s,ST
$s = s'$
return s

Thus, the complete approach of STEPS with Q-learning Algorithm is given as follows.

Algorithm 10 STEPS with Q-learning Algorithm

Input: n: Node initialized by Algorithm 1 and Algorithm 8

n set its state $s = Succ$
while True **do**
    n transmits by Aloha in uplink when there is a packet and open the downlink windows
    n evaluate the transmission and update the table with Algorithm 9 and set s with the return value
    for $sf \in [\![SF_{init}, 12]\!]$ **do** n calculates P_{sf} with Equation (6)
    end for
    n surfs among the parameters and choose a strategy based on the P_{sf} calculated.
end while

Note that Q-learning is a value-based algorithm. The choice of SF is based on values in the score table. Thus, the table is not normalized as in previous algorithms. However, this means the node only focuses on the SF with the highest score in the table. Although the ϵ-Greedy mechanism is implemented, the exploration is still not optimal since all SF without the highest value have the same probability. Therefore, the policy-based learning idea is investigated and implemented in the following subsection.

3) STEPS with Q-learning and policy-based Hybrid Algorithm:

The Q-learning and policy-based Hybrid algorithm is proposed by combining the Q-learning algorithm and policy-based learning idea. In addition, the variant of STEPS with Q-learning and policy-based Hybrid algorithm is also proposed. In this variant, the score table and the Q-table are separated. Agents do the same evaluation and update approaches as Q-learning to the Q-table. Then, agents generate a distribution of probabilities (i.e., policy) as the score table using the Q-table. Agents make strategic decisions based not on the highest value but on the probabilities in the score table (policy-based).

Considering that there may be negative values in the Q-table with Algorithm 9, the normalization in previous variants of STEPS is no longer suitable. Thus, a Softmax function is used as shown in Equation (10).

$$ST(s, sf) = \frac{exp\left(Q(s, sf)\right)}{\sum\limits_{sf \in [\![SF_{init}, 12]\!]} exp\left(Q(s, sf)\right)} sf \in [\![SF_{init}, 12]\!] \tag{10}$$

After integrating Equation (10) to Q-learning, the initialization of the score table is shown as in Algorithm 11. The example of the initialized Q-table and score table when $SF_{init}=9$ is as follows:

Q(s,a)	SF7	SF8	SF9	SF10	SF11	SF12
Succ	0	0	8	4	2	1
UL-Fail	0	0	8	4	2	1
Only-received	0	0	8	4	2	1

score	SF7	SF8	SF9	SF10	SF11	SF12
Succ	0	0	0.978	0.018	0.002	0.001
UL-Fail	0	0	0.978	0.018	0.002	0.001
Only-received	0	0	0.978	0.018	0.002	0.001

Algorithm 12 shows the evaluation and update phase, and the full approach of STEPS with Q-learning and policy-based Hybrid Algorithm is shown in 13.

Algorithm 11 Initial score table of Q-learning and policy-based Hybrid Algorithm

Input: SF_{init}: The initial SF chosen,
Output: ST: The initial score table,Q:The initial Q-table

Set $Q = \{\{0\}^{12}_{SF_{init}}, \{0\}^{12}_{SF_{init}}, \{0\}^{12}_{SF_{init}}\}$
Set $ST = \{\{0\}^{12}_{SF_{init}}, \{0\}^{12}_{SF_{init}}, \{0\}^{12}_{SF_{init}}\}$
for $state \in S$ **do**
    for $sf \in [\![SF_{init}, 12]\!]$ **do**
        Set $Q(state, sf) = 2^{12-sf}$
    end for
    for $sf \in [\![SF_{init}, 12]\!]$ **do**
        Set $ST(state, sf)$ with Equation (10)
    end for
end for
return ST,Q

To evaluate the performance of the proposed methods above, several simulations were carried out, the description of the simulations and the numerical results are detailed in next section.

V. Simulation and Evaluation

In this section, some simulations are carried out to verify the improvements of the proposed methods above. Since there are different types of agents, our simulator MULANE is used in this simulation work to have an objective experiment under the same environment.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

Algorithm 12 Evaluation and update phase with Q-learning and policy-based Hybrid Algorithm

Input: n: Current node, Q: Current score table of the node, sf: Current SF of the packet,SF_{init}: The initial SF chosen, c_a,c_r,c_f: The updating coefficients, s:current state
Output: s

The transmission is finished, n observe the ACK message for observed state o
if ACK received (i.e., $o = Succ$) **then**
    $s' = Succ$
else
    Calculate $prob$ with Equation (5)
    Let $s' = Only - received$ or $UL - Fail$ with a probability $prob$
end if
Get $r(s, a, s')$ with Equation (9) using s',c_a,c_r,c_f
Update $Q(s, a)$ with Equation (8) using s',a,s,Q
for $sf \in [\![SF_{init}, 12]\!]$ **do**
    Set $ST(state, sf)$ with Equation (10)
end for
$s = s'$
return s

Algorithm 13 STEPS with Q-learning Algorithm and policy-based Hybrid Algorithm

Input: n: Node initialized by Algorithm 1 and Algorithm 11

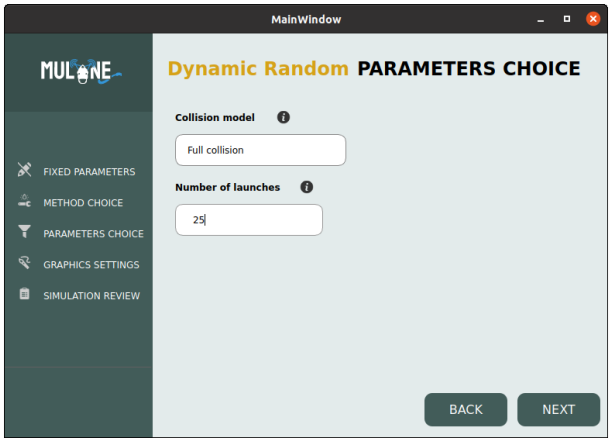
n set its state $s = Succ$
while True **do**
    n transmits by Aloha in uplink when there is a packet and open the downlink windows
    n evaluate the transmission and update the table with Algorithm 12 and set s with the return value
    n surfs among the parameters and choose a strategy based on the ST.
end while

A. Simulation Tool - MULANE

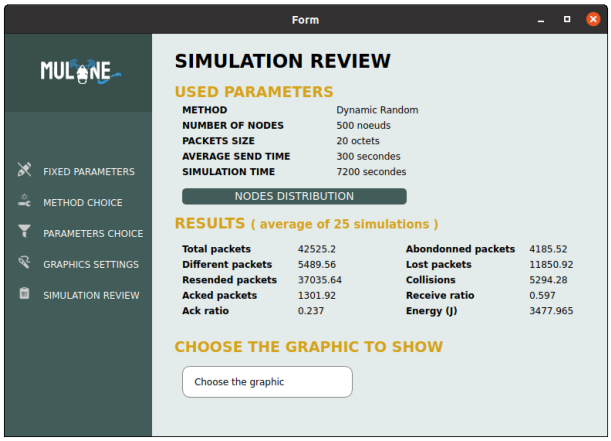
Before MULANE, different simulators were already used for evaluation in different studies. However, many of them have their hard-coded part, which is not flexible enough. Thus, combining the simulators or giving a unified simulation environment is challenging to compare the studies.

Facing this issue, we have developed MULANE in [36]. It is a lightweight agent-oriented simulator designed explicitly for LoRaWAN. By focusing on the agents' behaviors, MULANE is flexible and extendable for creating and simulating the devices with new algorithms. It is also easy to set up the environment with a .json file for different agents. Moreover, MULANE also provides a GUI interface to simplify the simulation and give the statistic results and data visualization, as shown in Fig. 6.

The following simulations consider the duty-cycle limitation for uplink and downlink and the full collision model. The other fixed parameters are given in Table I. In order to eliminate the accidental phenomena and reduce simulation errors, the program is run 25 times, and the average is taken as the final results for each simulation.



(a) Simulation parameters choosing



(b) Average results

Fig. 6: GUI interface of MULANE

TABLE I: Fixed Simulation Parameters

Parameters	Value
Channels	3 ISM 868Mhz band with 1% duty-cycle for both uplink and downlink, 1 channel with 10% duty-cycle for only downlink
Bandwidth	125kHz
Coding Rate	4/5
Retransmission times	8 Max.
MAC Header Length	1 byte
Payload Length	20 bytes
Simulation radius	5km
Packet rate	every exp(5 [mins])
Simulation Time	2 hours
Path Loss Model [7]	$\overline{Lpl(d0)} = 128.95dB$ $d0 = 1000m, \gamma = 2.32, \sigma = 7.8$
Base station number	1
Receiver sensitivities	See [37]
Receive Delay	1s for the first and 2s for the second
Transmission power	14dB
Collision Model	full collision model in [38]

B. Statistic Verification of Prior Knowledge Calculation

The first simulation is to check the $H_i(SF, d_i)$ calculated in Equation (5). Since the $H_i(SF, d_i)$ is an estimated value, a statistical verification is needed. With a normal distribution

model of the flat fading, the hypothesis to be verified can be presented as:

$\mathcal{H}$:The difference between $H_i(SF, d_i)$ calculated and the simulation result p_i of node i follows a 0-means Normal distribution. Where p_i represents the ratio that the packet is not lost.

By noting N_{tot} as the total packet the node sends, N_{lost} as the lost packet that the node suffers, and $K = 25$ the running times, the p_i can be calculated as:

$$p_i = \frac{\sum_K \frac{N_{tot} - N_{lost}}{N_{tot}}}{K} \tag{11}$$

The Student's t-test is used to verify the hypothesis $\mathcal{H}$. It is one of the most commonly applied statistical hypothesis tests that the test statistic follows a normal distribution [39]. The following are the procedures of the Student's t-test with n samples:

1) Calculate the statistic $x_i = H_i(SF, d_i) - p_i$. Note the mean of x_i as $\overline{x}$ and the standard deviation of x_i as σ
2) Calculate the confidence interval with Equation (12):

$$I = \left[\overline{x} - t_{\left(1-\frac{\alpha}{2}, n-1\right)} \frac{\sigma}{\sqrt{n}}, \overline{x} - t_{\left(1-\frac{\alpha}{2}, n-1\right)} \frac{\sigma}{\sqrt{n}}\right] \tag{12}$$

   Where $t_{\left(1-\frac{\alpha}{2}, n-1\right)}$ is the value of t-distribution with $n-1$ degrees of freedom and $100(1-\alpha)$ is the confidence coefficient.
3) If $0 \in I$, then the mean of x_i is not significantly different from 0 with a confidence level of $100(1-\alpha)$, which means the hypothesis $\mathcal{H}$ can be accepted. Otherwise, reject $\mathcal{H}$.

The statistical tests are made in a typical LoRaWAN scenario with 500 nodes distributed at equal intervals from 0 to 5000 meters away from the base station. All nodes use the same SF for transmission from 7 to 12. Fig. 7 shows the simulation and the Student's t-test results. The figures on the left column are the comparison between the calculated $H_i(SF, d_i)$ (Red dotted line) and the simulation results (blue points). It can be seen that for all SFs, the calculation results are located near the mean value of the simulation results. The higher SF gives a lower loss rate in transmission. Furthermore, it can be seen that with the increase of SF, the samples' variance goes higher. The reason may be that the high SF spends more time on the transmission, so the nodes transmit fewer packets and lead to more incertitude.

The right column of Fig. 7 shows the confidence intervals of each SF by Equation (12) with $\alpha = 0.1$ (the shadow part) and the average difference between calculated results and the simulation results. It can be seen that the 0 line (the red dotted line) always belongs to the intervals. This result means the simulation results are not significantly different from calculated $H_i(SF, d_i)$ with a confidence level of 90%, which means the hypothesis $\mathcal{H}$ can be accepted. The $H_i(SF, d_i)$ can estimate the transmission environment as prior knowledge.

It should be noted that the Student's t-test is used because 25 samples are less than 30. In the case of large simulation times, the Student's t-distribution is no longer suitable for validation [39]. In this case, chi-squared distribution and the G-test [40] is more suitable.

Table II shows some other indices of statistic results and the calculated results for each SF - The MAE (Mean Average Error), MSE (Mean Square Error), and the MAPE (Mean Average Percentage Error). It can be seen that for all SFs, all the indices are near 0. Thus, the calculated results with $H_i(SF, d_i)$ can be used as the prior knowledge to estimate the transmission.

TABLE II: Statistic Results

	MAE	**MSE**	**MAPE**
SF7	0.0265	0.0012	0.0770
SF8	0.0262	0.0012	0.0570
SF9	0.0256	0.0012	0.0434
SF10	0.0249	0.0013	0.0350
SF11	0.0254	0.0014	0.0317
SF12	0.0258	0.0015	0.0300

C. Performance of Prior-knowledge Initialization

In order to evaluate the performance of prior-knowledge initialization, several simulations are carried out using the variants of STEPS with the MDP problem mentioned above. These simulations replaced the initialization parts in selected methods with random initialization and compared them with proposed variants. For random initialization, $SF_{init} = 7$, the initial Q-table value is set as 0-arrays. Furthermore, since the initialization of SF is random, the mechanism of the downlink SF is also invalid. This means that the nodes without prior-knowledge initialization can only use the same SF as uplink transmission in downlink transmission.

The simulations are run in 50, 100, 250, 500, 600, 750 nodes. The following aspects were chosen as the performance indicators of the network:

- **Energy per 100 Ack**: is the energy consumed by the network for completing 100 successful bi-directional transmissions. This indicator represents the effective energy consumption in the network. The lower this value, the higher the network energy efficiency.
- **Ack Number**: is the total number of packets acknowledged by the base station. This indicator represents the throughput capacity of the network for confirmed message transmission.
- **ACK Ratio**: is the packets acknowledged ratio of the base station and the packets sent (the retransmissions are not included). It is the packet success rate (PSR) for bi-directional transmission. This indicator represents the capability of confirmed communications in the network.
- **Receive Ratio**: is the packets received ratio by the base station and the total transmission number, which includes the transmissions (includes the retransmission attempts). It is the packet delivery rate (PDR). Although packets are received by the base station but cannot send ACK because of reason 2 mentioned in Section I, this indicator can still represent the capability of avoiding collision and loss of the network since the transmissions successfully arrived at the destination.

For the parameters of Q-table, the reward parameters are taken as: $c_a = 10$, $c_r = -0.5$, $c_f = -2$. In the initialization

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

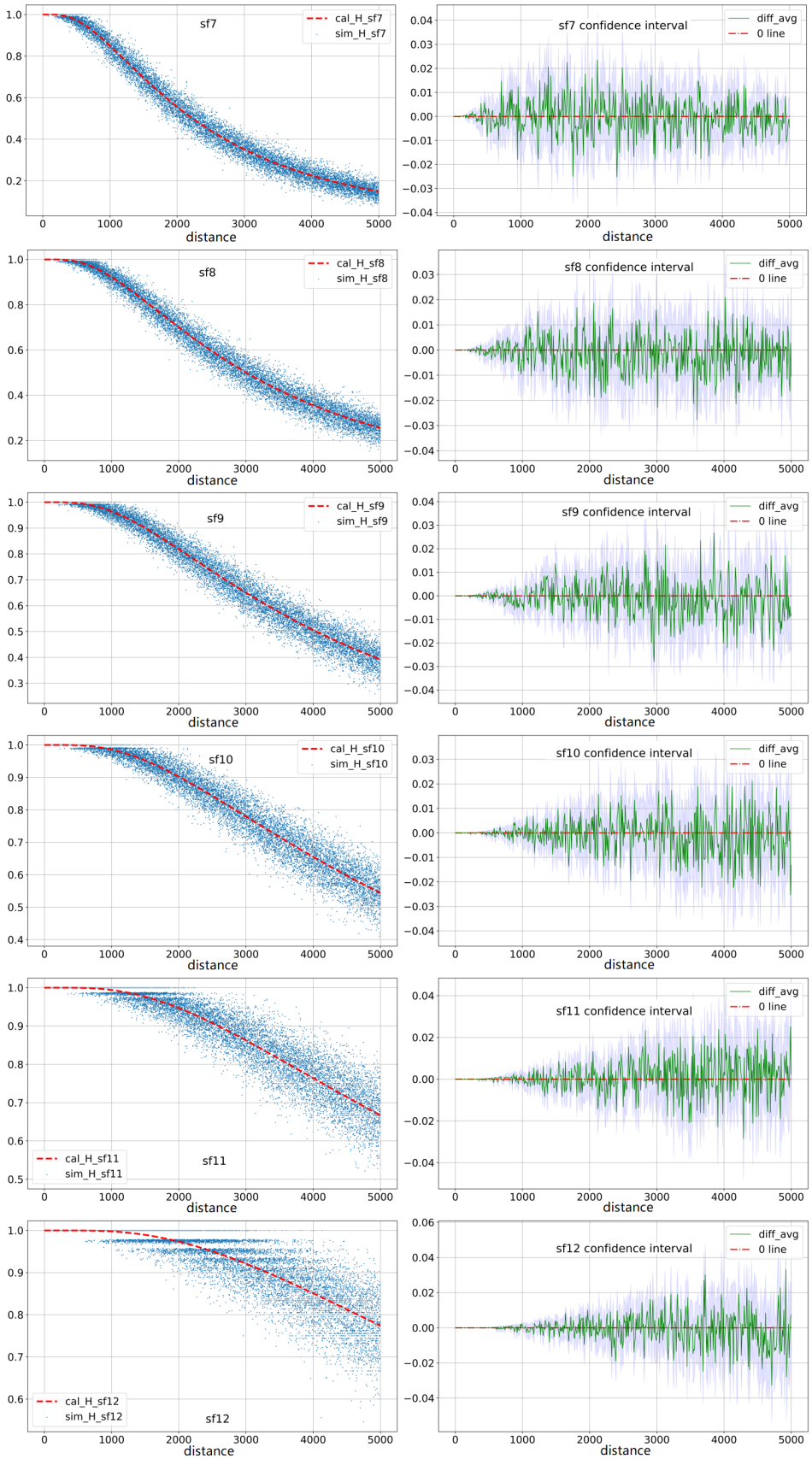


Fig. 7: Student's t-test results

part for prior-knowledge, $H_{thres} = 0.75$. For the exploration parameter in STEPS with Q-learning Algorithm, $\epsilon = 0.1$.

The simulation results are given in Fig. 8. In Fig. 8(a) and Fig. 8(b), it can be seen that the variants of STEPS with MDP problems can reduce 35% of energy consumption for a successful transmission while increasing 32% of the throughput with prior knowledge. Fig. 8(c) shows that the variants with prior knowledge have the highest Ack ratio, which is nearly 10% higher than those with random initialization. Fig. 8(d) shows that the nodes can avoid 12% more collisions or packet loss with prior knowledge.

All the figures in Fig. 8 demonstrate the improvement of the prior-knowledge initialization to the LoRaWAN network. Moreover, Fig. 8(a) and Fig. 8(b) show that STEPS with Q-learning and policy-based Hybrid Algorithm has a higher throughput than STEPS with Q-learning Algorithm with prior knowledge. This is probably because the hybrid algorithm makes better use of the Q-table and gives a better exploration strategy with the Softmax function given in Equation (10).

D. Comparisons of Different Methods

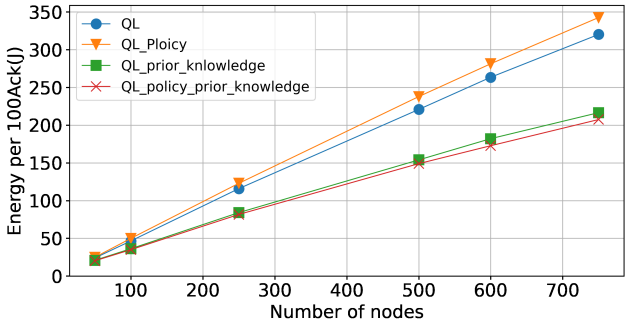
Several other simulations are carried out to show the benefits of the proposed algorithm. For comparisons, the nodes that have implemented the following dynamic methods are also simulated as the baseline methods in the same environment:

- **ADR**: The node chooses the initial SF randomly. The ADR runs at both node and server sides during transmission to allocate the LoRaWAN. The server-side approach is given in [16], and the ED-side algorithm is given in [17].
- **BADR**: The nodes allocate their SF with the method BADR in [21]. Nodes take turns selecting SF from the list [SF12, SF7, SF10, SF7, SF10, SF7] for the transmission.
- **LoRaMAB**: The node chooses the initial SF randomly. During the transmission, a MAB problem is modeled, and the EXP3.S algorithm is implemented to the node when choosing SF [25].

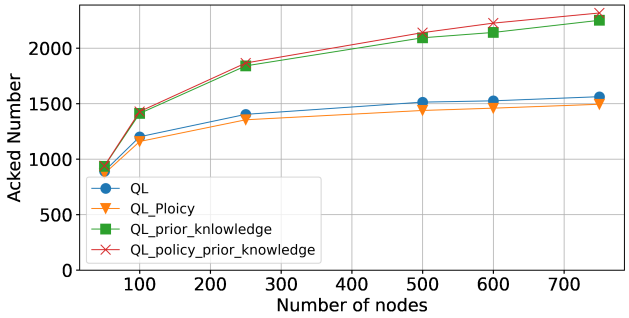
The codes of LoRaMAB are also open source in the repository of [41]. In order to have an objective experiment, the codes are modified and implemented into MULANE to ensure the same simulation environment.

The parameters of variants with MDP problems are kept as in the previous simulation. $p = 0.7$ for p-random surfing. For STEPS with prior-knowledge initialization and the variants with EE problems, $c_a = 3$, $c_r = 0.9$, $c_f = 0.8$, α is taken as 2. In the surfing phase, $\beta = 0.9$ for surfing phase with prior knowledge, $\epsilon = 0.1$ for ϵ-greedy and $\tau = 0.1$ in Boltzmann algorithm.

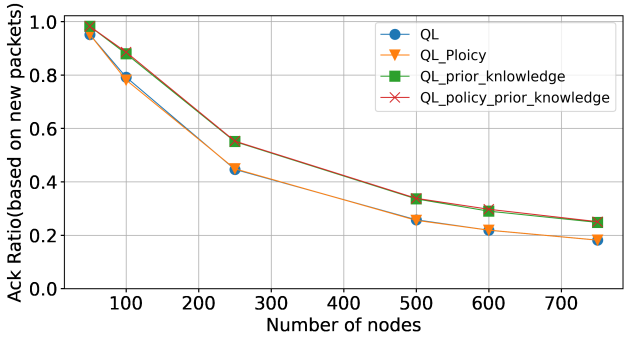
The simulations are first carried out with 100 nodes. The simulation results are given in Fig. 9. In Fig. 9(a) and Fig. 9(b), it can be seen that ADR spends the least energy for successful transmission. As a semi-centralized method, ADR allows the adjustment of SF at both sides and thus gets a high performance for bi-directional transmission. However, having a deterministic parameter selection policy, the server only decreases the SF in downlink messages. Thus, lots of devices are adjusted to the same SF and have a collision. This



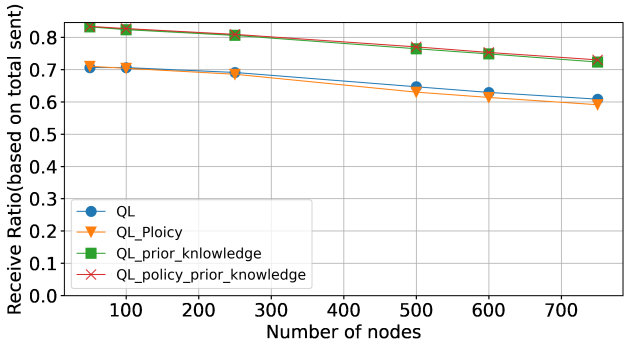
(a) Energy per 100 Ack



(b) Ack Number



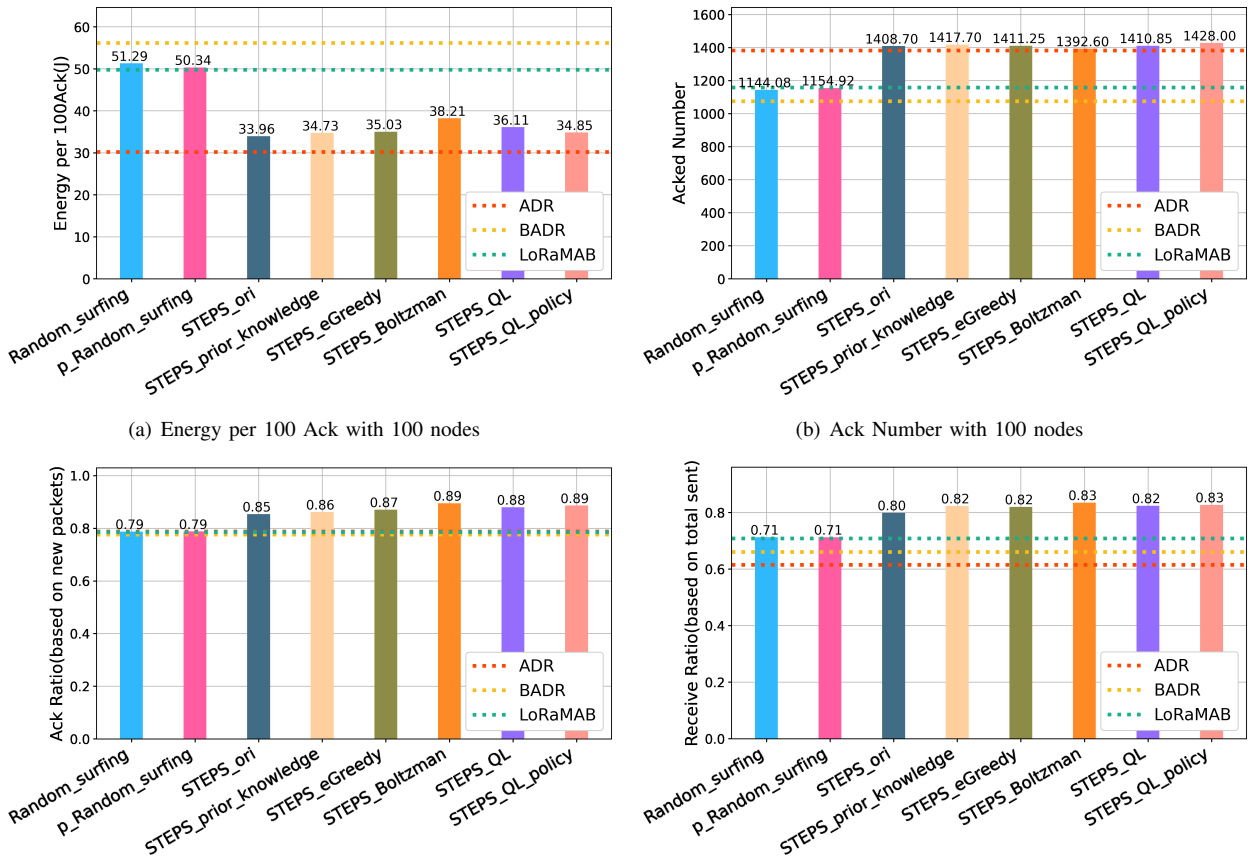
(c) Ack ratio (PSR)



(d) Receive ratio (PDR)

Fig. 8: Simulation Results and Comparison of prior-knowledge initialization

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301



(a) Energy per 100 Ack with 100 nodes

(b) Ack Number with 100 nodes

(c) Ack ratio (PSR) with 100 nodes

(d) Receive ratio (PDR) with 100 nodes

Fig. 9: Simulation Results and Comparison for 100 nodes

can be seen in Fig. 9(d) that ADR has the lowest PDR among all methods.

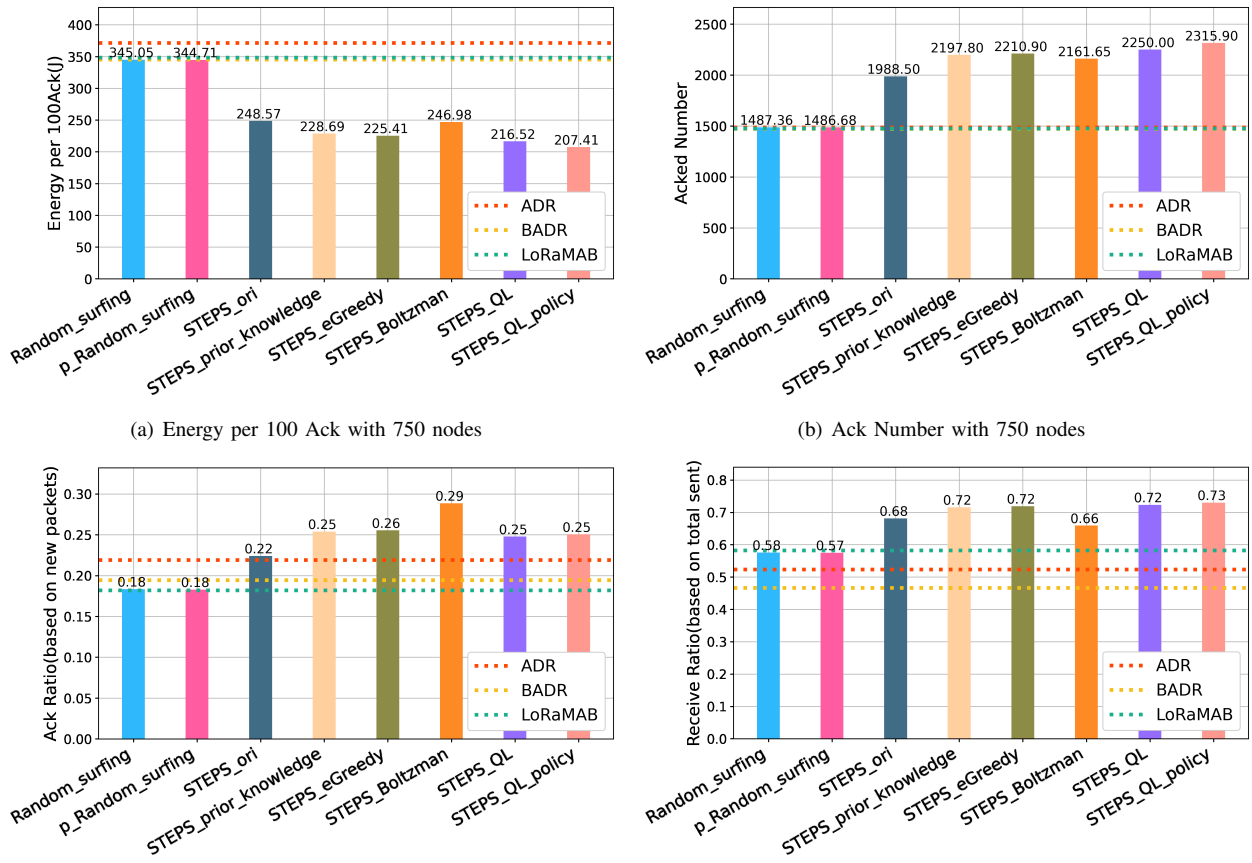
For the decentralized methods, it can be seen from Fig. 9(a) and Fig. 9(b) that the STEPS and its variants have a 18% higher throughput than other methods while decreasing 24% energy consumption. In Fig. 9(c) and Fig. 9(d), the STEPS with Q-Learning and the STEPS with Q-learning and policy-based Hybrid Algorithm have the highest PSR and PDR, which is 10% higher than ADR and BADR.

Furthermore, BADR, Random surfing, and p-Random surfing have low performance in all figures of Fig. 9. This demonstrates the importance of policy in decentralized methods. In BADR, the policy is deterministic, and in Random surfing and p-Random surfing, the policy is entirely random. With an inappropriate strategy and a lack of experience exploitation, nodes are likely to choose the inappropriate parameter and reduce network performance. The importance of the policy is further proved in the following simulations of ADR methods.

Fig. 10 shows the simulation results with 750 nodes in the network. In this case, the ACK resources at the gateway are exhausted because of the large number of nodes. Comparing Fig. 9(c) and Fig. 10(c), it can be seen that the ACK ratio significantly decreased. Thus, in ADR, nodes can not receive the downlink message and are stuck on the deterministic policy to choose a higher SF. A higher SF leads to a longer downlink transmission time which further occupies the downlink resources. Having this negative feedback, the energy consumption per success packet of ADR becomes the highest.

Compared with ADR, in Fig. 10 it can be seen that STEPS and its variants can still keep high performance in a large number of nodes. Fig. 10(a) and Fig. 10(b) show that the STEPS and its variants increase the network throughput by 33% at most while consuming 27% of energy for one success packet. Moreover, in Fig. 10(d), the proposed methods have a 10% higher PDR on average. This shows the anti-collision and anti-lost capability of the proposed methods.

All the figures in Fig. 9 and Fig. 10 demonstrate the improvement of the proposed methods for the LoRaWAN network. With previous experience in score tables, prior knowledge in initialization, and EE idea in strategy choosing or MDP modeling, STEPS and its variants bring a higher success rate, higher throughput capacity, and less energy consumption to the network. From the overall performance perspective, the variants of STEPS with MDP Problem give the best performance. STEPS with Q-learning and policy-based Hybrid Algorithm gives the highest throughput capacity and minor consumption in small and large node numbers. This shows that the STEPS framework can increase the successful communication rate of the network. However, compared with STEPS with prior knowledge, the STEPS with ϵ-Greedy Algorithm does not differ much. This may be caused by the parameter ϵ. For



(a) Energy per 100 Ack with 750 nodes

(b) Ack Number with 750 nodes

(c) Ack ratio (PSR) with 750 nodes

(d) Receive ratio (PDR) with 750 nodes

Fig. 10: Simulation Results and Comparison for 750 nodes

example, after the node has been transmitting for a while, it should know better the transmission environment, and the score table is more appropriate than the beginning. However, the node still gives a fixed amount of probability to explore, which wastes more opportunities for exploitation.

VI. Conclusion

This study investigates the impacts of parameter choosing on the LoRaWAN. By integrating the concept of reinforcement learning, some dynamic methods are proposed. Furthermore, using prior knowledge and modeling with different reinforcement learning algorithms, several variants of the STEPS method are proposed. The simulations showed that the prior knowledge calculation is statistically reliable, and the proposed methods can increase the success rate and the throughput capacity of the network with less energy consumption. For the future, there are some planned works:

- In STEPS with ϵ-Greedy Algorithm, the ϵ is always fixed, and enrichment of information is not used. In order to solve this problem, the dynamic value of ϵ may be implemented in the future.
- In [42], the authors proved the convergence of the Boltzmann exploration algorithm. However, the authors also proved that using the same or the monotonic learning rate for all strategies induces suboptimal behavior in some cases. Thus the authors proposed a method that uses different τ for different strategies. This can also be implemented in future works to improve the network further.
- In STEPS with MDP problem, the Q-table is established and updated based on experiences. The choice of parameters and reward functions can be further studied to improve performance.
- For the verification in a real network, it is also planned to build up a testbed to test the performance of proposed methods under a real LoRaWAN environment.

References

[1] LoRa Alliance Technical Committee *et al.*, "Lorawan® l2 1.0. 4 specification (ts001-1.0. 4)," *LoRa Alliance: Fremont, CA, USA*, 2020.

[2] LoRa Alliance Technical Committee *et al.*, "Rp002-1.0. 3 lorawan regional parameters," Tech. rep. Version: RP002-1.0. 3, Tech. Rep., 2021.

[3] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016.

[4] P. J. Basford, F. M. Bulot, M. Apetroaie-Cristea, S. J. Cox, and S. J. Ossont, "Lorawan for smart city iot deployments: A long term evaluation," *Sensors*, vol. 20, no. 3, p. 648, 2020.

[5] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[6] Semtech Corporation, "An in-depth look at lorawan class a devices," https://lora-developers.semtech.com/uploads/documents/files/LoRaWAN_Class_A_Devices_In_Depth_Downloadable.pdf, 2019 (accessed September, 2022).
[7] 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.
[8] G. Ferré, "Collision and packet loss analysis in a lorawan network," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2586–2590.
[9] F. H. Khan and M. Portmann, "Experimental evaluation of lorawan in ns-3," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2018, pp. 1–8.
[10] H. Rajab, T. Cinkler, and T. Bouguera, "Iot scheduling for higher throughput and lower transmission power," *Wireless Networks*, vol. 27, no. 3, pp. 1701–1714, 2021.
[11] M. Asad Ullah, J. Iqbal, A. Hoeller, R. D. Souza, and H. Alves, "K-means spreading factor allocation for large-scale lora networks," *Sensors*, vol. 19, no. 21, p. 4723, 2019.
[12] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "$free$fine-grained scheduling for reliable and energy-efficient data collection in lorawan," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 669–683, 2019.
[13] S. Narieda, T. Fujii, and K. Umebayashi, "Energy constrained optimization for spreading factor allocation in lorawan," *Sensors*, vol. 20, no. 16, p. 4417, 2020.
[14] Y. Kawamoto, R. Sasazawa, B. Mao, and N. Kato, "Multilayer virtual cell-based resource allocation in low-power wide-area networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 665–10 674, 2019.
[15] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of lorawans through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, 2018.
[16] Semtech Corporation, "Lorawansimple rate adaptation recommended algorithm," 2016.
[17] Semtech Corporation, "Understanding the lora® adaptive data rate," https://lora-developers.semtech.com/uploads/documents/files/Understanding_LoRa_Adaptive_Data_Rate_Downloadable.pdf, 2019 (accessed September, 2022).
[18] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "A fair adaptive data rate algorithm for lorawan," *arXiv preprint arXiv:1801.00522*, 2018.
[19] Y. A. Al-Gumaei, N. Aslam, X. Chen, M. Raza, and R. I. Ansari, "Optimising power allocation in lorawan iot applications," *IEEE Internet of Things Journal*, 2021.
[20] A. Farhad, D.-H. Kim, and J.-Y. Pyun, "R-arm: Retransmission-assisted resource management in lorawan for the internet of things," *IEEE Internet of Things Journal*, 2021.
[21] Semtech Corporation, "Lorawan® mobile applications: Blind adr," https://lora-developers.semtech.com/uploads/documents/files/LoRaWAN_Mobile_Apps-Blind_ADR_Downloadable.pdf, 2019 (accessed September, 2022).
[22] A. Kaburaki, K. Adachi, O. Takyu, M. Ohta, and T. Fujii, "Autonomous decentralized traffic control using q-learning in lpwan," *IEEE Access*, vol. 9, pp. 93 651–93 661, 2021.
[23] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, "Multi-armed bandit learning in iot networks: Learning helps even in non-stationary settings," in *International Conference on Cognitive Radio Oriented Wireless Networks*. Springer, 2017, pp. 173–185.
[24] A. Azari and C. Cavdar, "Self-organized low-power iot networks: A distributed learning approach," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.
[25] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "Lora-mab: Toward an intelligent resource allocation approach for lorawan," in *2019 IEEE global communications conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.
[26] M. Chen, L. Mokdad, J. B. Othman, and J.-M. Fourneau, "Steps-score table based evaluation and parameters surfing approach of lorawan," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.
[27] M. Chen, L. Mokdad, C. Charmois, J. Ben-Othman, and J.-M. Fourneau, "An mdp model-based initial strategy prediction method for lorawan," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4836–4841.
[28] O. Georgiou and U. Raza, "Low power wide area network analysis: Can lora scale?" *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, 2017.
[29] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan, and M. Gidlund, "Scalability analysis of a lora network under imperfect orthogonality," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1425–1436, 2018.
[30] M. R. Seye, B. Ngom, B. Gueye, and M. Diallo, "A study of lora coverage: range evaluation and channel attenuation model," in *2018 1st International Conference on Smart Cities and Communities (SCCIC)*. IEEE, 2018, pp. 1–4.
[31] P. Jörke, S. Böcker, F. Liedmann, and C. Wietfeld, "Urban channel models for smart city iot-networks based on empirical measurements of lora-links at 433 and 868 mhz," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–6.
[32] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of lpwans: range evaluation and channel attenuation model for lora technology," in *2015 14th international conference on its telecommunications (itst)*. IEEE, 2015, pp. 55–59.
[33] G. Callebaut and L. Van der Perre, "Characterization of lora point-to-point path loss: Measurement campaigns and modeling considering censored data," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1910–1918, 2019.
[34] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[35] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.
[36] M. Chen, L. Mokdad, J. B. Othman, and J.-M. Fourneau, "Mulane-a lightweight extendable agent-oriented lorawan simulator with gui," in *2021 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2021, pp. 1–6.
[37] E. D. Ayele, C. Hakkenberg, J. P. Meijers, K. Zhang, N. Meratnia, and P. J. Havinga, "Performance analysis of lora radio for an indoor iot applications," in *2017 International Conference on Internet of Things for the Global Community (IoTGC)*. IEEE, 2017, pp. 1–8.
[38] A. Rahmadhani and F. Kuipers, "Understanding collisions in a lorawan," *Surf Wiki*, 2017.
[39] J. F. Box, "Guinness, gosset, fisher, and small samples," *Statistical science*, pp. 45–52, 1987.
[40] J. H. McDonald, *Handbook of biological statistics*. sparky house publishing Baltimore, MD, 2009, vol. 2.
[41] T. Duc-Tuyen, "Loramab," https://github.com/tuyenta/IoT-MAB, 2019 (accessed Oct.2022).
[42] N. Cesa-Bianchi, C. Gentile, G. Lugosi, and G. Neu, "Boltzmann exploration done right," *Advances in neural information processing systems*, vol. 30, 2017.

Mi Chen Mi Chen received the B.S. degree in mathematics and applied mathematics and M.E. degree in industrial engineering at Beihang University (BUAA), China, in 2017 and 2020, respectively. He received the general engineer degree at Ecole Centrale de Lille, France in 2020. He is currently working toward the Ph.D. degree at the University Paris-Est, Créteil (UPEC) in Paris, France. His current research interests include the internet of things, IoT security, future wireless networks, and machine learning. He was a recipient of Best Paper award at GLOBECOM'21.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3239301

Lynda Mokdad received a PhD. in computer science from the University of Versailles, France in 1997. She was associate professor at University Paris-Dauphine from 1998 to 2009. She is currently full professor at University Paris-Est, Créteil since 2009.

Her main contributions focus on modeling, evaluation, quantitative verification, and security in the areas of wireless networks and Web services architectures. On the other hand, a cybersecurity axis was developed by focusing on denial-of-service issues in networks in order to counter them and vulnerabilities in the code for applications developed in C language on the other hand.

Jalel Ben-Othman received his B.Sc. and M.Sc. degrees both in Computer Science from the University of Pierre et Marie Curie, (Paris 6) France in 1992, and 1994 respectively. He received his PhD degree from the University of Versailles, France, in 1998. He is currently full professor at the University of Paris 13 since 2011 and member of L2S lab at CentraleSupélec. Dr. Ben-Othman's research interests are in the area of wireless ad hoc and sensor networks, VANETs, IoT, performance evaluation and security in wireless networks in general. He was the recipient of the IEEE COMSOC Communication Software technical committee Recognition Award in 2016, the IEEE computer society Meritorious Service Award in 2016, and he is a Golden Core Member of IEEE Computer Society, AHSN Exceptional Service and Contribution Award in 2018 and the VEHCOM Fabio Neri award in 2018. He has served as steering committee member of IEEE Transaction on Mobile computing (IEEE TMC), he is currently a senior Editor of IEEE communication letters (IEEE COMML) an editorial board member of several journals (IEEE Networks, IEEE IoT journal, IEEE TVT, JCN, IJCS, SPY, Sensors,). He has also served as TPC Co-Chair for IEEE Globecom and ICC conferences and other conferences as (WCNC, IWCMC, VTC, ComComAp, ICNC, WCSP, Q2SWinet, P2MNET, WLN....). He was the chair of the IEEE Ad Hoc and sensor networks technical committee January 2016-2018, he was previously the vice chair and secretary for this committee. He has been appointed as IEEE COMSOC distinguished lecturer from 2015 to 2018 and he is currently IEEE VTS distinguished lecturer where he did several tours all around the world.

Jean-Michel Fourneau is Professor of Computer Science at the University of Versailles St Quentin, France. He was formerly with Ecole Nationale des Telecommnications, Paris and University of Paris XI Orsay as an Assistant Professor. He graduated in Statistics and Economy from Ecole Nationale de la Statistique et de l'Administation Economique, Paris and he obtained is PHD and his habilitation in Computer Science at Paris XI Orsay in 87 and 91 respectively. He is an elected member of IFIP WG7.3 the IFIP working group on performance evaluation.

He is the Head of Computer Science Department at Versailles University and his recent research interests are algorithmic performance evaluation, Stochastic Automata Networks, G-networks, Energy Packet Networks, stochastic bounds, and application to high speed networks, and all optical networks.