IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2019, accepted October 2, 2019, date of publication October 17, 2019, date of current version October 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948111

Q-Learning Aided Resource Allocation and Environment Recognition in LoRaWAN With CSMA/CA

NAOKI AIHARA[1], KOICHI ADACHI[1], (Member, IEEE), OSAMU TAKYU[2], (Member, IEEE), MAI OHTA[3], (Member, IEEE), AND TAKEO FUJII[1], (Member, IEEE)

[1]Advanced Wireless and Communication Research Center, The University of Electro-Communications, Chofu 182-8585, Japan
[2]Department of Electrical and Computer Engineering, Shinshu University, Nagano 380-8553, Japan
[3]Department of Electronics and Computer Science, Fukuoka University, Fukuoka 814-0180, Japan

Corresponding author: Naoki Aihara (aihara@awcc.uec.ac.jp)

This work was supported by the MIC/SCOPE under Grant 175104004.

ABSTRACT The mutual interference among wireless nodes is a critical factor in the Internet-of-Things (IoT) era due to its dense deployment. Due to its large coverage area, wireless nodes may not be able to detect the on-going communication of other nodes in a long range wide area network (LoRaWAN), which is one of the low power wide area (LPWA) standards. This results in packet collision. The packet collision among LoRaWAN nodes significantly deteriorates network performance functions such as packet delivery rate (PDR). Furthermore, if packet collision happens, LoRaWAN nodes must retransmit packets, draining their limited battery power. Thus, mutual interference management among LoRaWAN nodes is important from the perspectives of both network performance and network lifetime. However, due to its large network size, it is difficult to explicitly comprehend the wireless channel environment around each LoRaWAN node, such as the relation among other LoRaWAN nodes. Thus, in this paper, we utilize the powerful machine learning technique. The wireless environment around LoRaWAN nodes are learned, and the knowledge is utilized for resource allocation in order to improve PDR performance. In the proposed method, Q-learning is adopted in a LoRaWAN system, and the weighted sum of the number of successfully received packets is treated as a Q-reward. The gateway (GW) allocates resources to maximize this Q-reward. The numerical results considering LoRaWAN elucidate that the proposed scheme can improve average PDR performance by about 20% compared to the random resource allocation scheme.

INDEX TERMS Frequency sharing, machine learning, resource allocation, LoRaWAN, CSMA/CA.

I. INTRODUCTION

To meet the demand for high speed communication, wireless access technologies have been evolving. Similarly, low power consumption communication is becoming more important despite the reduction of communication speed due to the emergence of the Internet-of-Things (IoT) [1]. Long range wide area network (LoRaWAN) is one of the promising network structures for low power wide area (LPWA) networks, which provide low speed, long range communication for distances of up to 10 km. Chirp-spread spectrum (CSS) technique is adopted for the physical layer of LoRaWAN. For the medium access control (MAC) layer, each LoRaWAN node adopts pure ALOHA. Due to this simple MAC protocol, increased packet collision due to the large number of LoRaWAN nodes is a critical factor in the limitation of the network performance. One of the countermeasures is the introduction of a duty cycle, which limits the transmission interval of each node to a predetermined threshold [2]. Recently, the application of carrier sense multiple access with collision avoidance (CSMA/CA) was proposed to improve the performance of LoRaWAN [3]. For example, CSMA/CA is essential for LoRaWAN in Japan [4]. In this protocol, LoRaWAN nodes detect the wireless medium before starting packet transmission. However, due to LoRaWAN's wide communication area and the low transmission power of its nodes, packet collision happens quite often in comparison to legacy wireless LAN systems. Because the LoRaWAN

The associate editor coordinating the review of this manuscript and approving it for publication was Kun Yang.

node has limited functionality due to its low cost, the introduction of more complicated interference management technologies into LoRaWAN nodes is not appropriate. One potential solution is to allocate orthogonal frequency channels to LoRaWAN nodes that often collide with each other. In LoRaWAN, there are up to 16 orthogonal frequency channels [5], and each LoRaWAN node randomly chooses one of the multiple available channels from information provided by its gateway (GW). However, it is difficult to decide which channel should be assigned to each LoRaWAN node due to the large scale of the network and the limited functionality of LoRaWAN nodes. Moreover, LoRaWAN nodes cannot inform the GW of the surrounding wireless environment, such as how often each LoRaWAN node can carrier sense (CS) the on-going communication due to its limited functionality. Thus, a resource allocation scheme that does not require such feedback from nodes is demanded. Conventional methods such as spreading factor (SF) allocation schemes are proposed for LoRaWAN in [7]–[9]. In [7], SF and coding rate are jointly assigned to ensure a high transmit success rate. Moreover, scalability [8] and coding rate fairness [9] are also considered. However, these works consider only ALOHA multiple access; no existing work considers orthogonal frequency channel allocation in LoRaWAN with CSMA/CA.

In this paper, we propose the utilization of a powerful machine learning technique for efficient orthogonal channel assignment in LoRaWAN with CSMA/CA. Because it is difficult to obtain the training set in a practical system, we focus on *reinforcement learning*, which does not require a training set but can learn the environment by observing the output from the environment after its action. To the best of our knowledge, this is the first work that tackles orthogonal resource allocation in LoRaWAN where additional information exchange is not allowed. The number of successfully received packets at a GW is used as the reward of learning so that no explicit feedback from a LoRaWAN node is needed for resource allocation. Because there is a strong correlation between the number of received packets and packet delivery rate (PDR), this resource allocation can improve PDR performance. The proposed scheme is shown to improve the average PDR performance by 20% compared to the random allocation scheme through a computer simulation with consideration for LoRaWAN specification.

The rest of this paper is organized as follows. In Sect. II, we introduce LoRaWAN and its system. In Sect. III, we summarize the system model considered in this paper. In Sect. IV, we briefly review the existing learning method. In Sect. V, we propose Q-learning based wireless resource allocation. In Sect. VI, computer simulation results are provided. Sect. VII concludes the paper.

II. LORAWAN

A. LORAWAN FUNCTIONS

1) PHYSICS LAYER

LoRaWAN is one of the LPWA standards, and adopts CSS modulation and frequency shift keying (FSK) as a physical layer technology. Its data rate and communication range are determined by SF. *SF* indicates the receive threshold. In a higher SF, the receiver can receive packets with lower received signal power, but the data rate from the transmitter is also reduced. Let the frequency bandwidth be W [Hz]. Then, the chip length T_c [sec] of the CSS symbol is given by $T_c = 1/W$ [sec]. The CSS symbol length T_s [sec] is given by

$$T_s = T_c \times 2^{SF}. \tag{1}$$

As *SF* increases, the transmitted signal has stronger resistance against noise and interference at the expense of the data rate. The typical data rate and signal-to-noise power ratio (SNR) limit is shown in Table 1.

TABLE 1. Data rate and SNR limit [10].

SF	Data rate [bps]	SNR limit [dB]
7	5469	-7.5
8	3125	-10
9	1758	-12.5
10	977	-15
11	537	-17.5
12	293	-20

The CSS modulated signal is transmitted over one of the orthogonal frequency channels. For LoRaWAN, there are up to K orthogonal frequency channels which depend on region and frequency [14]. Each GW informs the LoRaWAN nodes of the available channel indices [5].

2) MAC LAYER AND MULTIPLE ACCESS SCHEME

A simple ALOHA protocol is adopted as a MAC layer in LoRaWAN as its simple operation is suitable for low cost LoRaWAN nodes. Three classes are defined for LoRaWAN nodes, i.e., class A, B, and C [5]. Class A is mandatory for all LoRaWAN nodes. Class A nodes receive the downlink transmission together with an ACK message via two receive windows which are open after the uplink transmission of a node. Class B nodes periodically open a beacon receive window. Class C nodes always open a receive window. Thus, a GW can inform each node of necessary commands such as available frequency channels via downlink transmission.

III. SYSTEM MODEL

A. SYSTEM MODEL

Fig. 1 shows the LoRaWAN system considered in this paper. N LoRaWAN nodes are randomly and uniformly distributed within a network area of $D \times D$ [km^2]. One GW that controls LoRaWAN nodes and receives information from them is located at the center of the area. In total, K orthogonal frequency channels are available in this system. Let us denote the set of LoRaWAN nodes and that of the orthogonal frequency channels as $\mathcal{N}$ and $\mathcal{K}$, respectively.

Each LoRaWAN node generates packets of two different traffics [6]. The first traffic is regularly generated following predetermined packet generation interval $T_{\text{interval},n}$, which is selected from the set $\mathcal{T}_{\text{interval}}$. In this study, the LoRaWAN

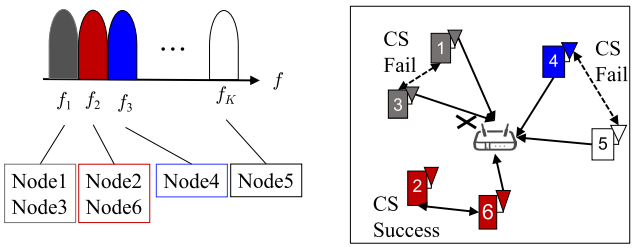


FIGURE 1. System model.

nodes that have the same packet generation intervals are called *cluster*. A random offset $T_{\text{offset},n} \sim \mathcal{U}[0, T_{\text{interval},n}]$ is assigned to LoRaWAN node n. The packet generation interval indicates the application type in the communication area such as gas meter and water supply meter. Note that even the LoRaWAN nodes with the same packet generation interval may not transmit the packets simultaneously owing to the different offsets. We assume that there are U application types. Thus, on an average, (N/U) LoRaWAN nodes generate packets with the same interval and attempt to send the packet to the GW. The second traffic is generated once an event is detected. In this study, an event (e.g., fire and electricity accident) occurs at time T_{event} in each epoch at a random position, and it propagates in the communication area with predetermined speed [6]. The exponential propagation model is considered in this study.

Each LoRaWAN node transmits packets in accordance with its duty cycle. After finishing a transmission of packet i, LoRaWAN node n waits for the transmission of packet $(i+1)$ transmission until $T_{\text{wait},n,i}$, which is given by

$$T_{\text{wait},n,i} = \frac{1-G}{G} \times \left(\lceil (N_{\text{trans},n,i}/R_b) \rceil \times T_s \right), \tag{2}$$

where $N_{\text{trans},n,i}$ is the transmitted packet size of packet i from node n, R_b is the data rate, and G is the duty cycle.

If multiple LoRaWAN nodes transmit packets to the GW using the same frequency channel simultaneously, the GW receives multiple packets. If both the SNR and the signal-to-interference power ratio (SIR) are above the thresholds Γ_{SNR} and Γ_{SIR}, respectively, the packet is considered to be successfully received. If the transmitted packet is lost, the LoRaWAN node retransmits packets based on binary-backoff [11]. The backoff length is calculated by uniform distribution with [0,CW], where CW is given as

$$\text{CW} = \text{CW}_{\min} \times 2^{N_r}, \tag{3}$$

where $\text{CW}_{\min}$ is the minimum backoff length, and N_r is the number of retransmissions.

B. CHANNEL MODEL

The received signal power of LoRaWAN node n at GW is given as

$$P_{\text{r},n}[\text{dBm}] = P_{\text{t},n}[\text{dBm}] - P_{\text{pathloss}}(d_n)[\text{dB}] - \psi[\text{dB}], \tag{4}$$

where $P_{\text{t},n}$ is transmit power of LoRaWAN node n, $P_{\text{pathloss}}(d_n)$ is a path loss component, ψ is shadowing component that is a function of location of LoRaWAN node (x_n, y_n). Pathloss is given as

$$P_{\text{pathloss}}(d_n) = 10\, a \log_{10} d_n + b + 10\, c \log_{10} f_{\text{c}}, \tag{5}$$

where d_n is the distance between LoRaWAN node n and the GW [km], and f_{c} is the carrier frequency [MHz]. Propagation parameters a, b, and c are the coefficients for distance, offset, and frequency loss component, respectively. For the propagation model between LoRaWAN nodes, we adopt the same model given by (4) and (5) with different parameters [13] because GWs are generally located above LoRaWAN nodes.

C. PROBLEM FORMULATION

Let us denote the PDR of LoWaWAN node n by P_n^{del}, which is given by

$$P_n^{\text{del}} = \frac{R_n}{S_n}, \tag{6}$$

where R_n denotes the number of successfully received packets from LoRaWAN node n while S_n denotes the number of packets generated during a predetermined time length T_{epoch}. Hereafter, this time length is defined as *epoch*.

The optimal channel selection aims to choose a proper channel to maximize the expected PDR of each node, i.e.,

$$k_n^{\star} = \arg\max_{k_n \in \mathcal{K}} \mathbb{E}\left[P_n^{\text{del}} \right], \tag{7}$$

where $\mathbb{E}[x]$ denotes the ensemble average operation.

In this study, channel allocation is executed every epoch. In this model, R_n depends on the channel allocation of other nodes due to their interference. This makes the optimization problem one of combination optimization, i.e. it cannot be solved in practical time. Moreover, S_n also depends on other system parameters.. For example, a large T_{wait} i.e. small duty cycle G makes the number of transmitted packets small. Thus, the number of successfully received packets becomes small. However, interference also becomes smaller as network traffic is reduced. This phenomenon also happens in the case of large CW.

At GW, it is not possible to know S_n without explicit feedback from LoRaWAN node n. To solve this problem, we propose reinforcement learning-based optimization and approximation of the objective function using only the number of successfully received packets.

IV. MACHINE LEARNING TECHNOLOGY

A. Q-LEARNING

Reinforcement learning is one of the learning schemes that search for optimal action from a given situation. The agent is not given the pair of the specific situation with the optimal action. However, the agent is given the reward for a specific situation and a corresponding action. The agent executes the optimal action based on a reward function that is the sum of the reward of each action. However, this reward function

depends on the environment, and it is difficult to solve this function theoretically. To tackle this, the agent approximates the reward function from a taken action and a given reward. This learning scheme is efficient for the specific situations where actions affect subsequent situations, or for situations which provide results from a series of actions, e.g. Markov chain.

1) Q-LEARNING MODEL

Let $\mathcal{S}$ and $\mathcal{A}$ be a state set and an action set, respectively. Then, the reward function at time t is approximated by the following update equation:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(S_{t+1}, a) - Q(S_t, A_t)\right], \tag{8}$$

where $Q(S_t, A_t)$ is the expected value of a reward when an agent takes action $A_t \in \mathcal{A}$ in state $S_t \in \mathcal{S}$. R_{t+1} is an instant reward at time $t+1$ of action A_t, $\gamma \in [0, 1]$ is a discount rate, and $\alpha \in (0, 1]$ is a Q-learning rate. This approximation converges to real reward function ($Q^\star$) [16].

2) Q-LEARNING USING NEURAL NETWORK

In traditional Q-learning, the agent needs to keep the Q values for each pair of state and action by using eq. (8); therefore, the agent keeps it as a table of state and action because the Q-value is calculated for each combination of state and action. This format requires an enormous memory capacity when the number of combinations of state and action ($|\mathcal{S}| \times |\mathcal{A}|$) is large. For example, in this study, $|\mathcal{S}|$ exponentially increases because it is expressed by the combination of resources allocated to each LoRaWAN node, i.e., $|\mathcal{S}| = |\mathcal{K}|^{|\mathcal{N}|}$ and $|\mathcal{A}|$ linearly increase,s with the number of available frequency channels, i.e., $|\mathcal{A}| = |\mathcal{K}|$. To avoid such a large memory capacity requirement , approximating Q values using a neural network (NN) is proposed [17], which is called Deep-Q-Network (DQN). An agent can get the approximate model for input and output functions by giving pairs of input and output to NN. By using this, NN learns the weights to output a Q-value approximation for each action $A_t \in \mathcal{A}$ from input of the current state $S_t \in \mathcal{S}$, as shown in Fig.2.

B. NEURAL NETWORK (NN)

NN is one of machine-learning schemes that approximate the relationship between input and output information using neurons [18]. This learning scheme contains two steps: forward propagation and back propagation, as shown in Fig.3. In this research, the GW has N NNs for each of N LoRaWAN nodes. In this section, without loss of generality, we review the NN function of node n.

1) FORWARD PROPAGATION

NN is composed of neurons and couplings. Each neuron is arranged hierarchically, and has two functions: reception and activation, as shown in Fig. 4. First, a neuron obtains the

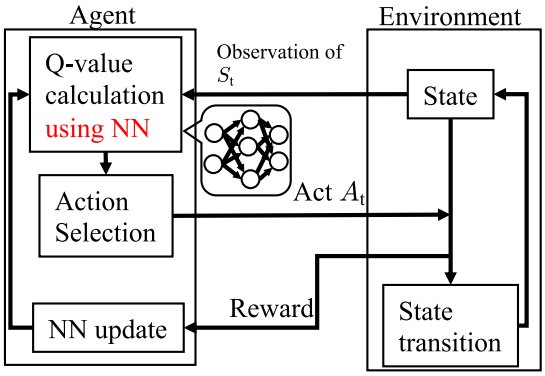


FIGURE 2. Model of Q-learning based on NN.

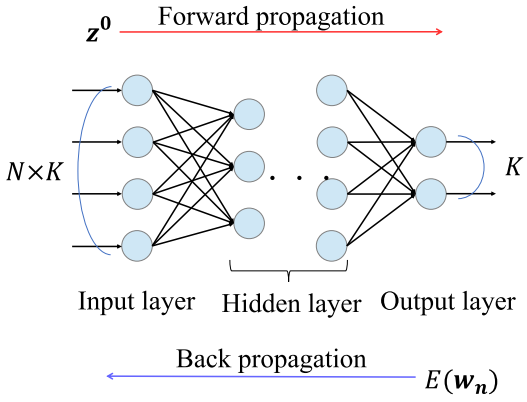


FIGURE 3. Forward propagation and back propagation.

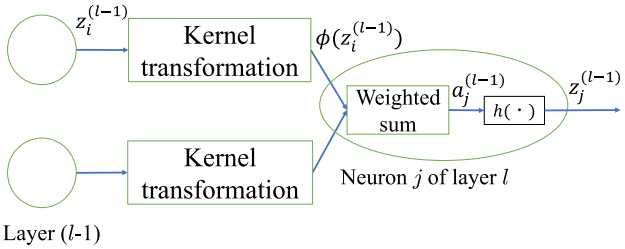


FIGURE 4. Calculation model of NN.

weighted sum of output from the previous layer. Then, the neuron transforms it by an activation function that is generally nonlinear. Neuron j of layer l receives the weighted sum of the output of neurons in layer $l-1$ as

$$a_{n,j}^{(l)}\left(\mathbf{z}_n^{(l-1)}, \mathbf{w}_{n,j}^{(l-1)}\right) = \sum_i w_{n,i,j}^{(l-1)} \phi(z_{n,i}^{(l-1)}), \tag{9}$$

where $w_{n,i,j}^{(l-1)}$ is the coupling weight from neuron i of layer $l-1$ to neuron j of layer l, $z_{n,i}^{(l-1)}$ is the output of neuron i in layer $l-1$, and $\phi(x)$ is a kernel function. For the kernel function, an ideal function is applied as

$$\phi(x) = x. \tag{10}$$

Then, neuron j in hidden layer l calculates output $z_{n,j}^{(l)}$ by applying an activation function to $z_{n,j}^{(l)}$ as

$$z_{n,j}^{(l)} = f(a_{n,j}^{(l)}), \tag{11}$$

where $f(\cdot)$ is the activation function. Generally, rectified linear unit (ReLU) function f_{ReLU} is used for the activation function, which is given as

$$f_{\mathrm{ReLU}}(x) = \max(0, x). \tag{12}$$

2) BACK PROPAGATION

The NN weights, $\mathbf{w}_n = \{w^l_{n,i,j}\}$, are trained by using the error function between the output from the NN and the desired output. Let the error function for given NN weights $\mathbf{w}_n$ be $E(\mathbf{w}_n)$. Then, the optimal NN weights, $\mathbf{w}_{n,\mathrm{opt}}$, satisfy

$$\nabla E(\mathbf{w}_{n,\mathrm{opt}}) = 0, \tag{13}$$

where ∇ is a gradient operator. However, since it is hard to derive the optimal weights analytically, it is common to derive it using numerical scheme. The NN parameters are updated from a learning time τ as

$$\mathbf{w}_n^{\tau+1} = \mathbf{w}_n^{\tau} - \Delta_n^{\tau}, \tag{14}$$

where Δ_n^{τ} is an update term at τ , and initial weights $\mathbf{w}_n^0$ are calculated using Xavier initialization [21].

There are many methods that calculate update term Δ_n^{τ}, such as stochastic gradient descent (SGD), adaptive moment estimation (Adam), etc. The gradient value of each coupling weight, $\frac{\partial E_n}{\partial w^{(l)}_{n,i,j}}$, is required to calculate Δ_n^{τ}. Back propagation (BP) is an efficient method to calculate the gradient. Let us focus on the update of weight $w^{(l)}_{n,i,j}$. The gradient is calculated as

$$\frac{\partial E_n}{\partial w^{(l)}_{n,i,j}} \leftarrow \delta^l_{n,j} z^{(l)}_{n,i}, \tag{15}$$

where $\delta^l_{n,i}$ is called the error gradient that is expressed as

$$\delta^l_{n,j} = \begin{cases} \dfrac{\partial E_n}{\partial z^{(l+1)}_{n,j}} \dfrac{\partial z^{(l+1)}_{n,j}}{\partial a^{(l+1)}_{n,j}} & \text{if } l = L-2 \\ \dfrac{\partial f(a^{(l+1)}_{n,j})}{\partial a^{(l+1)}_{n,j}} \sum_k w^{(l+2)}_{n,j,k} \delta^{(l+1)}_{n,k} & \text{otherwise,} \end{cases} \tag{16}$$

where L is the number of layers of NN. In this paper, a squared error is adopted as the error function $E(\mathbf{w})$, which is given by

$$E(\mathbf{w}_n) = \frac{1}{2}(o_{n,k} - z^{(L-1)}_{n,k})^2, \tag{17}$$

where $o_{n,k}$ is the training data, and $z^{(L-1)}_{n,k}$ is the approximation of the training data with output k. In this study, we want to approximate Q-function, i.e., $o_{n,k} = Q_{n,k_n}$ where Q_{n,k_n} is actual Q-reward when resource k_n is allocated to LoRaWAN node n.

3) OPTIMIZER

NN parameter $\mathbf{w}$ is updated as shown in (14) using the gradient value as described above. For calculating Δ_n^{τ}, there are several schemes such as SGD and Adam. In SGD, the gradient value is directly used to calculate and update Δ_n^{τ}. Although SGD can escape from a local optimal point, it has the slowest convergence. In Adam, the 1st moment of the gradient is normalized by the 2nd moment of the gradient in order to adapt the learning rate and stabilize calculation. By this normalization, the parameter fluctuation can be suppressed.

For NN, a pure perceptron with L layers is adopted in this paper. Let us define layer 0 as the input layer and layer $(L-1)$ as the output layer, and the other layers are defined as hidden layers. The update equation for NN weights depends on the optimizer. Let us describe the weight update between neuron i of layer l and neuron j of layer $l+1$ for the LoRaWAN node n. In SGD, weight $w^{(l)}_{n,i,j}$ is updated by

$$w^{(l)}_{n,i,j} \leftarrow w^{(l)}_{n,i,j} + \eta \times \frac{\partial E_n}{\partial w^{(l)}_{n,i,j}}, \tag{18}$$

where η is an NN learning rate.

In Adam, the update equation is given by

$$w^{(l)}_{n,i,j} \leftarrow w^{(l)}_{n,i,j} + \eta \times \frac{\hat{m}^{(l)}_{n,i,j,t}}{\sqrt{\hat{v}^{(l)}_{n,i,j,t}} + \epsilon_{\mathrm{Adam}}}, \tag{19}$$

where $\hat{m}^{(l)}_{n,i,j,t}$ is the estimated 1st moment of the gradient at epoch t, $\hat{v}^{(l)}_{n,j,k,t}$ is the estimated 2nd moment of the gradient, η is the learning rate, and ϵ_{Adam} is a small value to avoid division by zero. $\hat{m}^{(l)}_{n,j,k,t}$ and $\hat{v}^{(l)}_{n,j,k,t}$ are given by the below equations:

$$\hat{m}^{(l)}_{n,i,j,t} = \frac{m^{(l)}_{n,i,j,t}}{1-\beta_1^t} \tag{20}$$

$$\hat{v}^{(l)}_{n,i,j,t} = \frac{v^{(l)}_{n,i,j,t}}{1-\beta_2^t}, \tag{21}$$

where

$$m^{(l)}_{n,i,j,t} = \begin{cases} (1-\beta_1)\dfrac{\partial E_n}{\partial w^{(l)}_{n,i,j}} & \text{if } t = 0 \\ \beta_1 m^{(l)}_{n,i,j,t-1} + (1-\beta_1)\dfrac{\partial E_n}{\partial w^{(l)}_{n,i,j}} & \text{otherwise,} \end{cases} \tag{22}$$

$$v^{(l)}_{n,i,j,t} = \begin{cases} (1-\beta_2)\left(\dfrac{\partial E_n}{\partial w^{(l)}_{n,i,j}}\right)^2 & \text{if } t = 0 \\ \beta_2 v^{(l)}_{n,i,j,t-1} + (1-\beta_2)\left(\dfrac{\partial E_n}{\partial w^{(l)}_{n,i,j}}\right)^2 & \text{otherwise.} \end{cases} \tag{23}$$

V. PROPOSED SCHEME

A. DESIGN OF LEARNING MODEL

Let *one epoch* be composed of the channel allocation, Q value observation, and learning process. The GW has one independent Q-learning equipment for each LoRaWAN node , i.e.,

GW acts as an agent of Q-learning. The frequency channel assignment for the next epoch is determined based on the state at the current epoch. Without loss of generality, we explain the frequency channel assignment for LoRaWAN node n. Let us define state set $\mathcal{S}$, action set $\mathcal{A}$, and Q-value as below.

- State $\mathcal{S}$: The combination of the allocated channel indices of all the nodes. The frequency channel assignment for each LoRaWAN node is represented by one-hot-K vector, where the element corresponding to the assigned frequency channel is set to 1, and otherwise set to 0. Thus, each state $S_t \in \mathcal{S}$ is a column vector by stacking up N one-hot-K vector. For example, suppose $N = 3$ and $K = 4$ and then one possible state is given by $(1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)$.
- Action $\mathcal{A}$: The set of channel indices which can be allocated to node n.
- Q-reward Q_{n,k_n}: The weighted sum of the number of received packets. It is adjusted by the ratio of the number of received packets from the node n of interest and the minimum number of received packets of other nodes, which is given as

$$Q_{n,k_n} = D_{n,t+1} + \nu \times \frac{\sum_{n' \in \mathcal{N} \setminus n} D_{n',t+1}}{N-1}, \tag{24}$$

where $D_{n,t}$ is the number of successfully received packets from LoRaWAN node n during epoch t, and ν is a *selfish parameter* that adjusts priority between node n's reward and that of other nodes. The selfish parameter ν is expressed as

$$\nu = \tanh\left(\frac{D_{n,t+1}}{\min_{n' \in \mathcal{N} \setminus n} D_{n',t+1}}\right). \tag{25}$$

When ν is small, node n acts selfishly and tries to increase its own number of received packets. On the other hand, if ν is large, GW attempts to equalize the performance of all nodes through resource allocation to node n.

This learning contains a two step learning comprised of wireless environmental learning and optimal resource selection. The first part learns the wireless environment around each LoRaWAN node from the input channel allocation state. For example, this learning tries to understand which pair of LoRaWAN nodes do not interfere with each other even if they are allocated to the same frequency channel. The second part is frequency channel allocation based on the wireless environment. Based on the learned wireless environment, the optimal frequency channel is assigned to each LoRaWAN node. In the proposed scheme, the two steps are connected and the frequency channel allocation is performed based on the input frequency channel allocation state.

B. RESOURCE ALLOCATION USING Q-LEARNING

The GW allocates one of K frequency channels to each LoRaWAN node based on the output from NN as follows. We show the allocation algorithm at epoch t.

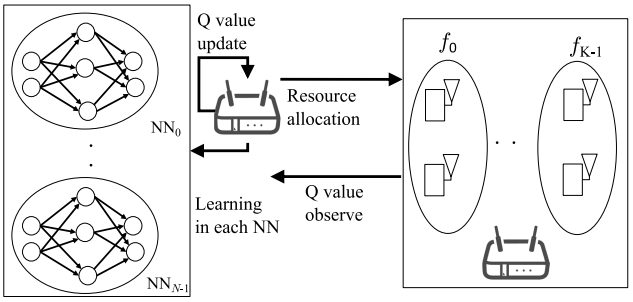


FIGURE 5. Proposed model.

Step1 The agent inputs the current resource allocation state S_t to the NN of each LoRaWAN node and obtains output Q-value Q_{n,k_n} from each NN.

Step2 With probability $\epsilon(t)$, the GW randomly allocates one of K frequency channels to LoRaWAN node n. With probability ($1 - \epsilon$ (t)), the GW allocates frequency channel $k_n^\star$ to LoRaWAN node n, where $k_n^\star$ is given by

$$k_n^\star = \arg\max_{k_n \in \mathcal{K}} Q_{n,k_n}(S_t). \tag{26}$$

Step3 The GW observes the number of successfully received packets for state S_t.

By this, the GW can allocate the frequency channel that maximizes the number of successfully received packets having a correlation with the PDR to each LoRaWAN node.

VI. SIMULATION RESULTS

A. SIMULATION PARAMETERS AND MODEL

In this section, we provide computer simulation results to verify the performance of the proposed scheme. In this simulation, node-GW shadowing is calculated by a spatially correlated shadowing model [12]. This component is expressed as a function of the location of node $\psi(x_n, y_n)$. Between LoRaWAN nodes, shadowing is calculated using uncorrelated shadowing because the nodes are located near ground height, and the shadowing correlation is very low due to the distance between nodes. This component is therefore expressed as the function of nodes index $\psi(n, q)$ where n and q are indices of nodes. In both situations, uncorrelated shadowing is based on log-normally distributed shadowing loss with zero-mean and standard deviation of σ [dB]. The wireless system parameters and the learning parameters are summarized in Tables 2 and 3, respectively. LoRaWAN's parameter is derived from the Japanese parameter configuration AS923 from document [14]. For learning parameters, we compare learning schemes and models, e.g. the number of layers, learning rate, optimizers, activate functions, etc. The optimal combination of learning parameters and schemes is used for PDR performance evaluation. For the ϵ-greedy scheme, $\epsilon(t)$ is given as

$$\epsilon(t) = \frac{T - t}{T}, \tag{27}$$

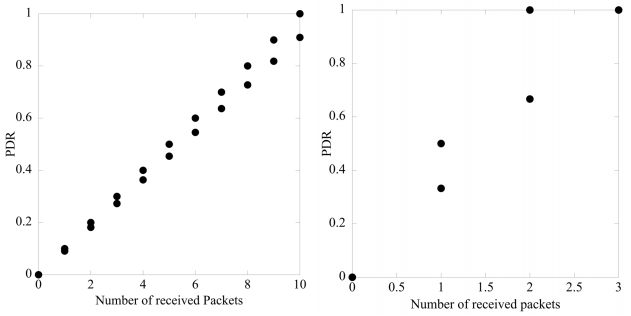
where t is the current epoch, and T is the number of epochs.

TABLE 2. Wireless system parameter.

Simulation area $D \times D$	3×3 [km^2]
Spreading factor SF	12
Bandwidth W	125 [kHz]
Symbol time T_s	32.768 [ms]
Duty cycle G	0.01
Transmit power P_t	13 [dBm]
Carrier frequency f_c	923 [MHz]
Number of LoRaWAN nodes N	500
Propagation coefficient of distance loss a	(2.0, 4.0)
Propagation offset b	(32.45, 9.5)
Propagation coefficient of frequency loss c	(2.0, 4.5)
Shadowing deviation σ	3.48 [dB]
Shadowing correlation coefficient	0.05
Noise power spectrum density	-174 [dBm/Hz]
Noise figure	9 [dB]
Packet sizeN_t	240 [bits]
Event occur time T_{event}	5 [min]
Packet generation interval set $\mathcal{T}_{\text{interval}}$	{60, 300} [sec]
Probability of packet generation cluster	{0.5,0.5}
Event occur time in each epoch	300 [sec]
Event propagation speed	700 [m/sec]
Event propagation coefficient	0.005
CS threshold Γ_{CS}	{-80, -90, -100, -110} [dB]
Receive threshold SNR Γ_{SNR}	-20 [dB] [15]
Capture effect threshold SIR Γ_{SIR}	6.0 [dB] [15]
Number of frequency channels K	{2, 4, 8, 16}

TABLE 3. Learning parameter.

Optimizer	SGD [18],Adam [19]
activation function	ReLU
Length of one epoch for learning	600 [s]
Number of epochs for learning T	500
Q-learning rate α	0.4
Number of NN layers L	3, 4, 5
Number of neurons of hidden layer	10 ($L = 3$), (10, 5) ($L = 4$), (10, 5, 5) ($L = 5$)
Adam's parameter $(\beta_1, \beta_2, \epsilon_{\text{Adam}})$	(0.9, 0.999, 10^{-8})



(a) Correlation of Cluster 0. (b) Correlation of Cluster 1.

FIGURE 6. Correlation between the number of the number of received packets and PDR.

B. CORRELATION OF PDR AND THE NUMBER OF RECEIVED PACKETS

We first validate the use of the number of received packets for PDR improvement. Fig. 6 shows the average PDR performance as a function of the number of received packets. In order to quantitatively show the correlation between these two metrics, the Pearson product moment correlation coefficient [20] is calculated as

$$\rho = \frac{\sum_{t=0}^{T-1}\sum_{n=0}^{N-1}(R_{n,t} - \overline{R})(P_{\text{del},n,t} - \overline{P_{\text{del}}})}{\sqrt{\sum_{t=0}^{T-1}\sum_{n=0}^{N-1}(R_{n,t} - \overline{R})^2)\sum_{t=0}^{T-1}\sum_{n=0}^{N-1}(P_{\text{del},n,t} - \overline{P_{\text{del}}})^2)}}, \tag{28}$$

where $\overline{R}$ and $\overline{P_{\text{del}}}$ are average values of PDR and the number of received packets. The Pearson correlation coefficient ρ^2 is approximately 0.95 for cluster 0 and 0.70 for cluster 1 for this setup. For other parameter setups, the following cases are considered

Case 1 $N = 500$ and $\mathcal{T}_{\text{interval}} = \{120\}$ [sec]
Case 2 $N = 1000$ and $\mathcal{T}_{\text{interval}} = \{60, 300\}$ [sec] with uniform probability.

For Case 1, ρ^2 is approximately 0.86. For Case 2, ρ^2 is approximately 0.97 for cluster 0 (i.e., $T_{\text{interval},n} = 60$ [sec]), and approximately 0.84 for cluster 1 (i.e., $T_{\text{interval},n} = 300$ [sec]). Thus, these results indicate that the number of successfully received packets and the PDR have a strong correlation, justifying the use of the number of successfully received packets instead of PDR.

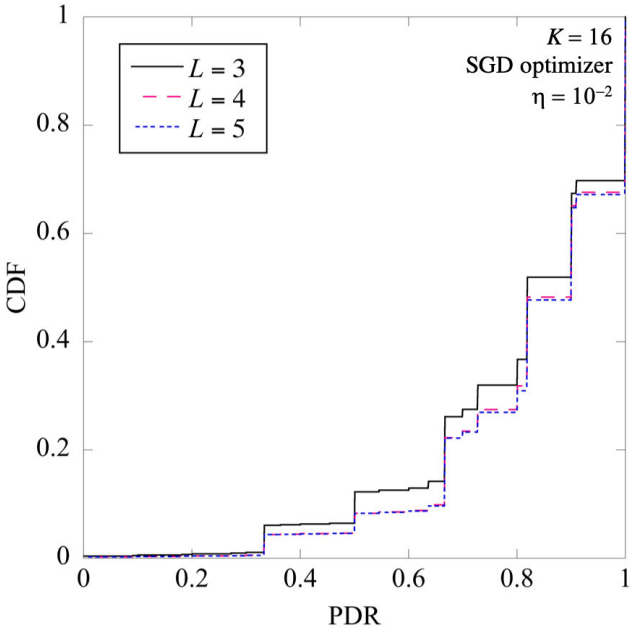


FIGURE 7. Impact of the number of layers, *L*.

C. COMPARISON OF LEARNING SCHEME

1) NUMBER OF LAYERS

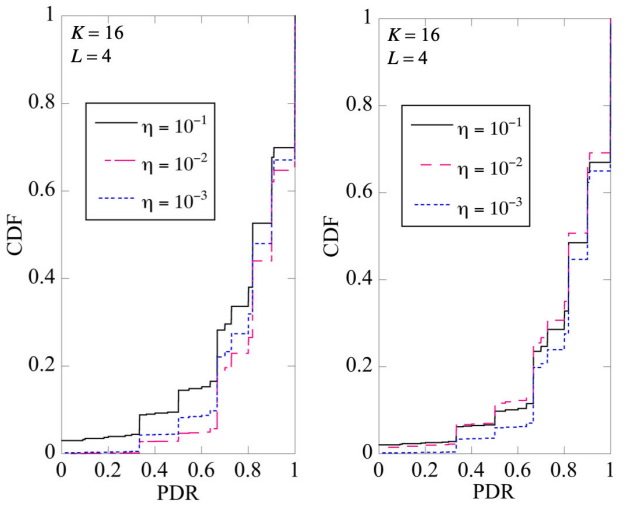
It is well known that the number of layers L has a strong impact on the output value of NN. Fig. 7 shows the impact of L on the PDR performance when ReLU activation is used for the activation function. SGD is adopted as an optimizer.

As shown in Fig. 7, the optimal number of layers is 4 for the ReLU activation function. This can be explained as follows. If the number of layers is too small, the performance of the NN is insufficient to express the relationship. On the other hand, if the number of layers is too large, the so called vanishing gradient problem [21] occurs. In other words, the gradient of the error function approaches zero, so the NN cannot be trained. This problem becomes more obvious as the number of layers increases. Moreover, the initial state is not good when the number of layer is large. However, the NN can be effectively trained when the gradient of error function is sufficiently large. Thus, an appropriate number of layers exists. In the following evaluation, $L = 4$ is used for NN with the ReLU activation function.

The reason for the stair-like curve is as follows. Because the PDR is evaluated at each epoch as shown in eq. (6), the maximum number of packets to be received is at most $T_{\text{epoch}}/T_{\text{interval},n} + 1 = 11$. Thus, for example, the PDR performance takes an integer multiple of $1/11$.

2) LEARNING RATE

Next, the impact of the optimizer on the PDR performance is shown for each activation function.



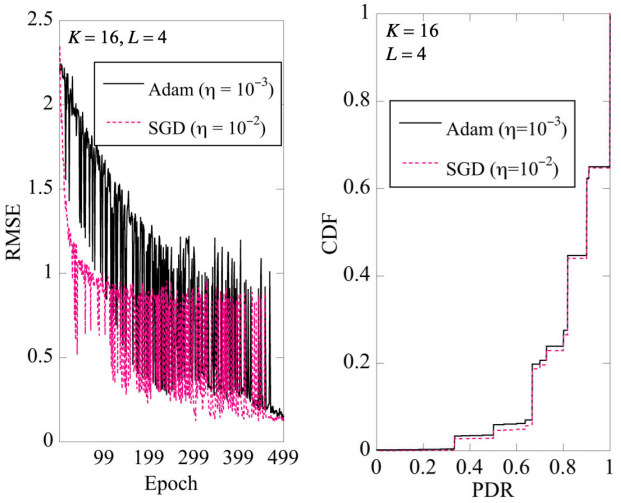
(a) SGD optimizer. (b) Adam optimizer.

FIGURE 8. Impact of learning rate, η.

Figure 8 shows the PDR performance when the SGD optimizer and the Adam optimizer are used. For the SGD optimizer, learning rate $\eta = 10^{-2}$ shows the best PDR performance, while $\eta = 10^{-3}$ provides the best performance when the Adam optimizer is used.

3) LEARNING SCHEME

Based on the optimum values for the number of layers L and the learning rate η for each optimizer, the root mean squared error (RMSE) convergence property and the PDR performance of each learning scheme is shown in Fig. 9.



(a) RMSE convergence. (b) CDF of PDR.

FIGURE 9. Comparison of learning schemes.

The RMSE is defined as

$$RMSE = \frac{\sum_{n=0}^{N-1}(Q_{n,k_n} - z_{n,k}^{(L-1)})^2}{N}. \tag{29}$$

Fig. 9a shows that, although the RMSE becomes smaller as learning progresses, it does not converge to 0. This is because, in this learning model, each node may change the resource index used in epoch $t + 1$ from that in epoch t; thus, the Q-value is not always stable on state S_t. This result shows that the SGD optimizer can become small faster than the Adam optimizer. If the Adam optimizer is used, the latter data sets have relatively small effects relative to the former. This results in the latter part having worse convergence performance. Fig. 9b shows the CDF of PDR for different combinations of activation function and optimizer. Although the difference between the performances of two optimizers is relatively small, the computational complexity of Adam is greater than that of SGD. This is because Adam requires additional computations such as the square root of the second moment. Thus, in the following evaluation, SGD optimizer is used.

D. PDR PERFORMANCE

Fig. 10 shows how the learning proceeds. It can be seen from the figure that the PDR value improves as learning progresses.

The impacts of number of frequency channels K and CS threshold Γ_{CS} on the CDF of PDR performance are shown in Fig. 11 and Fig. 12. Fig. 11 shows that the performance improvement from the proposed scheme depends on the number of available frequency channels, K. When $K = 8$, the proposed scheme can improve the average PDR performance by 20%, compared with the conventional random allocation. However, when $K = 16$, this performance improvement becomes slightly smaller and is approximately 13%. This is

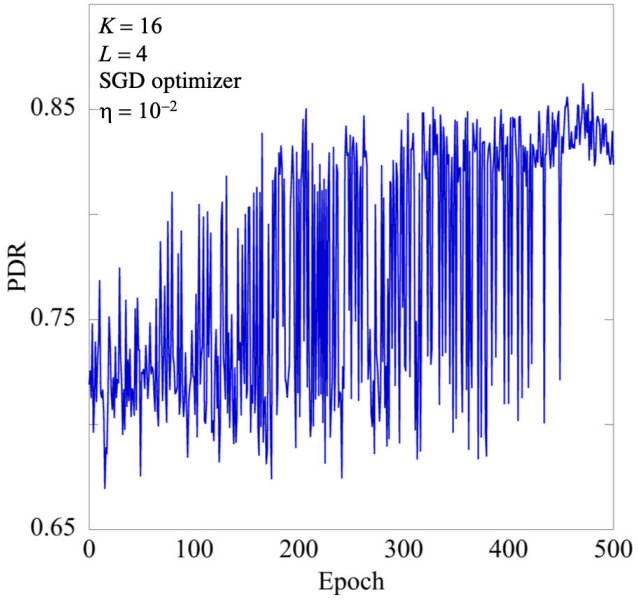


FIGURE 10. Process of learning.

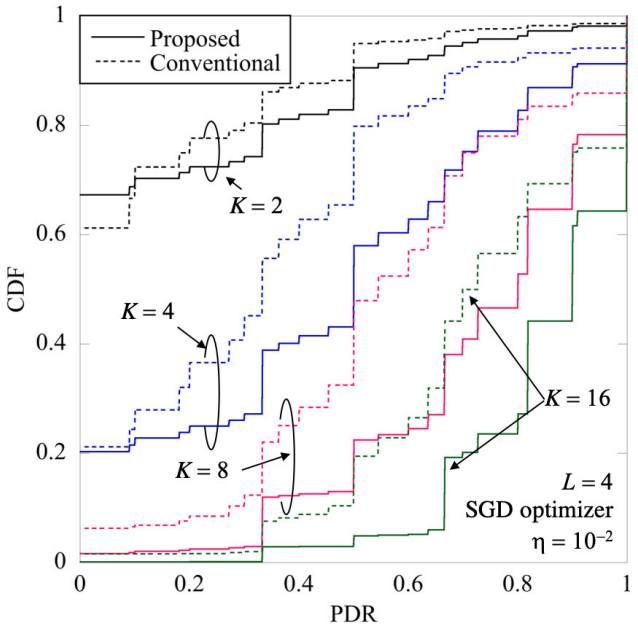


FIGURE 11. Impact of the number of resources, K.

because random channel hopping can avoid packet collision if the system has a sufficient number of channels. In a typical LoRaWAN system, only a small number of channels is available. For example, in the EU standard, the minimum number of channels is set to 3 [6]. Thus, it can be said that the proposed scheme is more effective in a practical environment. From the figure, it can be seen that there are sharp transitions of performance at PDR close to 1 and 0 (this is more obvious for a small number of K). These phenomena can be explained as follows. First, the reason for the sharp transition close to PDR = 1 is due to the nodes close to a GW. Because such nodes are close to GW, the received signal power at

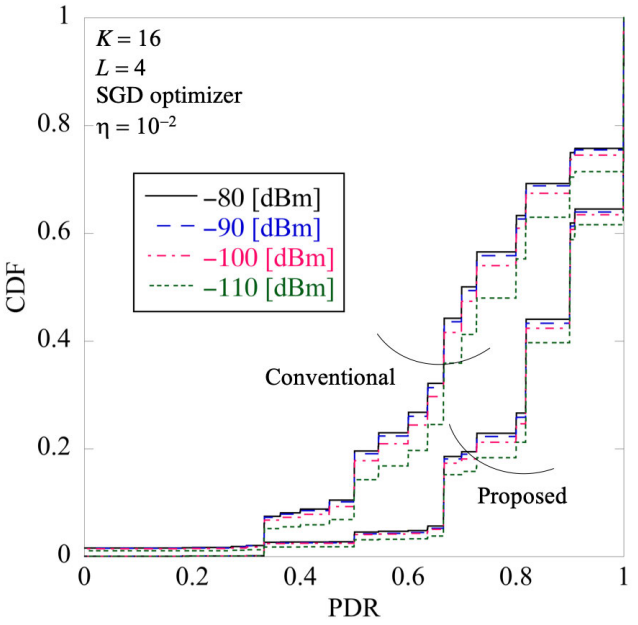


FIGURE 12. Impact of CS threshold, Γ_{CS}.

GW is considerably high; therefore, their PDR performances is almost 1 even under interference. Second, the reason for the sharp transition close to PDR = 0 is due to the interference among the LoRaWAN nodes. Even with the proposed scheme, strong mutual interference among the nodes may occur. If the mutually interfering LoRaWAN nodes are assigned to the same frequency channel, they interfere with each other and result in packet loss. If the number of available frequency channel K is small, this interference cannot be avoided by random channel hopping.

Because the PDR performance of LoRaWAN with CSMA/CA highly depends on how accurately each LoRaWAN node can CS with each other, we evaluate the impact of the CS threshold Γ_{CS}. Fig. 12 shows that as Γ_{CS} becomes lower, the conventional random channel allocation can slightly improve the PDR performance by avoiding packet collision. However, even when Γ_{CS} is low, packet collisions still happen due to the correlation of the packet generation timing. On the other hand, the proposed scheme can provide much better performance irrespective of Γ_{CS} because the proposed scheme can allocate frequency channels to avoid packet collision depending on hidden terminal relations and the correlation of packet generation. The proposed scheme can avoid allocating identical channel frequencies to nodes that either have the same packet generation timing or cannot CS each other. Through the learning, the proposed scheme, having higher priority, avoids one of the two factors that significantly impact the PDR performance degradation.

E. WIRELESS ENVIRONMENT RECOGNITION

There are two factors that result in packet collision: CS availability and packet transmission timing collision.

To evaluate those factors, we define three metrics, i) CS rate P_{CS}, ii) packet transmission timing difference $T_{\mathrm{PG}}(n,q)$, and iii) mean packet transmission timing difference in each frequency channel $\overline{T}_{\mathrm{PG}}(k)$ with $k \in \mathcal{K}$. First, P_{CS} is defined as

$$P_{\mathrm{CS}} \triangleq \frac{\sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{C}(k)}\sum_{q\in\mathcal{C}(k)} I_{\mathrm{CS}}(n,q)}{\sum_{k\in\mathcal{K}}\sum_{n\in\mathcal{C}(k)}\sum_{q\in\mathcal{C}(k)} 1}, \tag{30}$$

where k is the frequency channel index, n and q are the node indices, $\mathcal{C}(k)$ is the set of LoRaWAN nodes allocated to the frequency channel k, $I_{\mathrm{CS}}(n,q)$ is the indicator functions given by

$$I_{\mathrm{CS}}(n,q) = \begin{cases} 1, & \text{if node } n \text{ and node } q \text{ can CS each other} \\ 0, & \text{otherwise} \end{cases}. \tag{31}$$

$T_{\mathrm{PG}}(n,q)$, and $\overline{T}_{\mathrm{PG}}(k)$ are defined as

$$T_{\mathrm{PG}}(n,q) = \min_{i,j} |t_{n_i} - t_{q_j}| \text{ s.t.} n, q \in \mathcal{C}(k), \tag{32}$$

$$\overline{T}_{\mathrm{PG}}(k) = \mathbb{E}_{n,q}[T_{\mathrm{PG}}(n,q)], \tag{33}$$

where t_{n_i} is the approximated starting time of transmission of packet i from node n. This is given by

$$t_{n_i} = T_{\mathrm{Offset},n} + i \times \delta_{\mathrm{packet},n}, \tag{34}$$

where $\delta_{\mathrm{packet},n}$ is the interval between packet $i-1$ and i, which takes into account the duty cycle T_{wait} as $\delta_{\mathrm{packet},n} = \max\left(T_{\mathrm{interval},n}, T_{\mathrm{wait},n,i}\right)$. Because GW also shall follow the duty cycle [22]. packet retransmission is not allowed. Thus, the error between the actual packet transmission starting time and the approximated one is negligible, i.e., on the order of contention window.

If the CS rate P_{CS} is high, the LoRaWAN nodes allocated to the same frequency channel can CS each other; hence, packet collision can be avoided. If the packet generation timing difference $T_{\mathrm{PG}}(n,q)$ is large, the LoRaWAN nodes allocated to the same frequency channel can also avoid packet collision. Thus, from the view point of wireless environment recognition and frequency channel allocation, it is desirable to have high P_{CS} or large $T_{\mathrm{PG}}(n,q)$.

1) CS RATE

The CS rate, P_{CS}, of random allocation and the proposed scheme are shown in Fig. 13 The figure shows that the proposed scheme improves P_{CS} slightly, compared with random allocation. This is because the packet generation timing difference is more dominant than CS availability. In the following, we show this.

2) PACKET TRANSMISSION STARTING TIME

The CDF performances of packet transmission with starting time difference $T_{\mathrm{PG}}(n,q)$ of the proposed scheme and the conventional random allocation scheme are plotted in Fig. 14. For reference, the performance of the system with $K = 1$ is also plotted. As Fig. 14a shows, the proposed scheme can slightly increase the value compared with the random allocation by allocating frequency channels such that the LoRaWAN nodes in the same frequency channel have a greater time difference. Fig. 14b shows the mean value of the packet transmission starting time difference, $\overline{T}_{\mathrm{PG}}(k)$. This result shows that the variance of mean difference is greater for the proposed scheme. Although the improvement is marginal, the PDR performance is significantly improved. From these results, it is indicated that the proposed scheme can increase the mean of difference, while maintain the required transmission time difference.

This trend becomes obvious if the packet generation timing offsets of LoRaWAN nodes are close to each other. We assume the case that $T_{\mathrm{Offset},n}$ is randomly selected from six predetermined values. If the LoRaWAN nodes in the

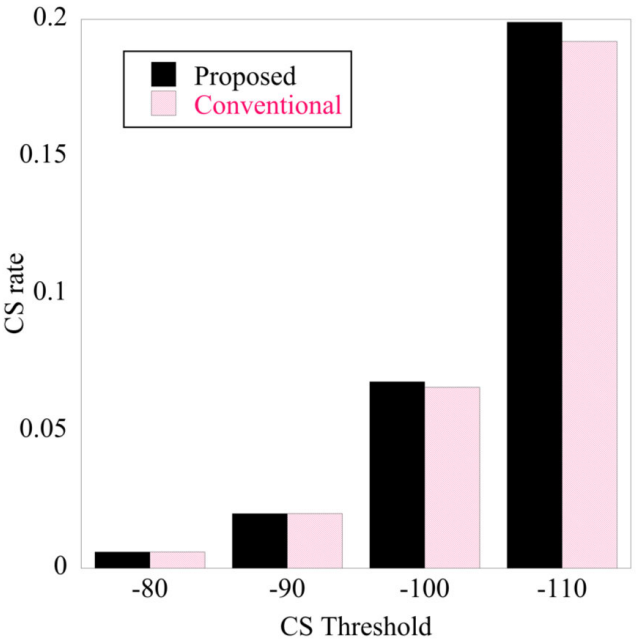
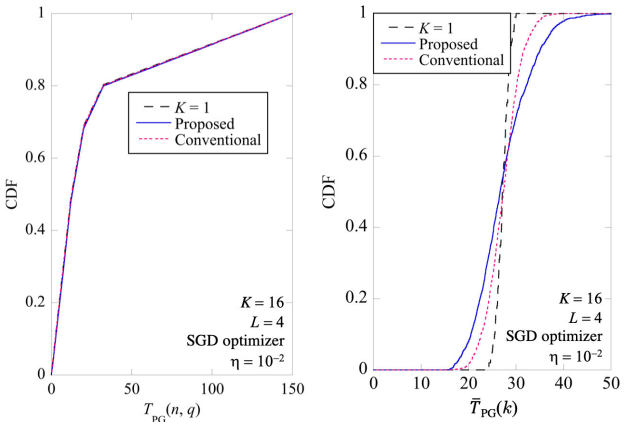
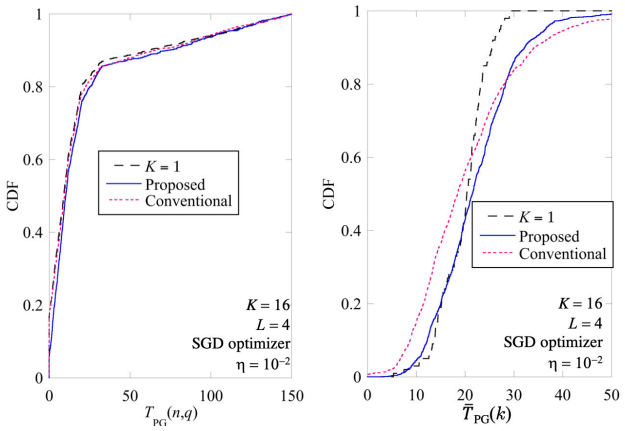


FIGURE 13. CS Ratio.



(a) CDF of $T_{\mathrm{PG}}(n,q)$. (b) CDF of $\overline{T}_{\mathrm{PG}}(k)$.

FIGURE 14. Performance of packet transmission starting timing difference.



(a) CDF of $T_{\mathrm{PG}}(n, q)$. (b) CDF of $\overline{T}_{\mathrm{PG}}(k)$.

FIGURE 15. Performance of packet transmission starting timing difference when timing offset is clustered.

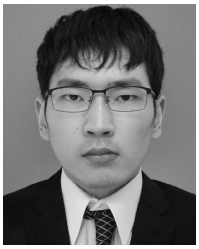
same frequency channel have the same $T_{\mathrm{Offset},n}$, then we have $T_{\mathrm{PG}}(n, q) = 0$. As Fig. 15 shows, the probability having $T_{\mathrm{PG}}(n, q) = 0$ can be significantly lowered by the proposed scheme compared with random allocation. Furthermore, $\overline{T}_{\mathrm{PG}}(k)$ of the proposed scheme is 3 [sec] greater than that of random channel allocation. Thus, the proposed scheme can effectively avoid packet collision among LoRaWAN nodes in the same frequency channel.

VII. CONCLUSION

In this paper, we have proposed a wireless resource allocation scheme to avoid mutual interference from hidden nodes in CSMA/CA and from traffic collision, and we have evaluated this scheme using computer simulation. By searching for an optimal resource allocation that can maximize the weighted sum of the number of successfully received packets from each node using Q-learning and NN approximation, each node can avoid packet collision without explicit feedback. From computer simulation, it is shown that the proposed scheme can improve the average PDR performance by about 20%. These results indicate that the proposed method could improve packet delivery performance without preparing more wireless resources and explicit feedback that drain LoRaWAN node batteries.

REFERENCES

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, 2nd Quart., 2017. doi: 10.1109/COMST.2017.2652320.
[2] D. Tian, J. Zhu, and Z. Nie, "An improved LoRaWAN protocol based on adaptive duty cycle," in *Proc. IEEE 3rd Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Chongqing, China, Oct. 2017, pp. 1122–1125.
[3] J. Ortin, M. Cesana, and A. Redondi, "How do ALOHA and listen before talk coexist in LoRaWAN?" in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, Sep. 2018, pp. 1–7.
[4] *920 MHz-Band Telemeter, Telecontrol and Data Transmission Radio Equipment*, Standard T108, 2012.
[5] LoRa Alliance. *LoRaWAN 1.1 Specification*. Accessed: May 24, 2019. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf
[6] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT traffic and its impact on LoRaWAN," in *Proc. IEEE GLOBECOM*, Singapore, Dec. 2017, pp. 1–6.
[7] M. El-Aasser, T. Elshabrawy, and M. Ashour, "Joint spreading factor and coding rate assignment in LoRaWAN networks," in *Proc. IEEE Global Conf. Internet Things (GCIoT)*, Alexandria, Egypt, Dec. 2018, pp. 1–7.
[8] A. Tiurlikova, N. Stepanov, and K. Mikhaylov, "Method of assigning spreading factor to improve the scalability of the LoRaWan wide area network," in *Proc. 10th Int. Congr. Ultra Mod. Telecommun. Control Syst. Workshops (ICUMT)*, Moscow, Russia, Nov. 2018, pp. 1–4.
[9] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Fair adaptive data rate allocation and power control in LoRaWAN," in *Proc. IEEE 19th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Chania, Greece, Jun. 2018, pp. 14–15.
[10] A. Lavric and V. Popa, "Internet of Things and LoRa low-power wide-area networks: A survey," in *Proc. Int. Symp. Signals, Circuits Syst. (ISSCS)*, Lasi, Romania, Jul. 2017, pp. 1–4.
[11] C. D. M. Cordeiro and D. P. Agrawal, *Ad Hoc and Sensor Networks: Theory and Applications*. Singapore: World Scientific, 2006.
[12] H. Claussen, "Efficient modelling of channel maps with correlated shadow fading in mobile radio systems," in *Proc. IEEE 16th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Berlin, Germany, Sep. 2005, pp. 512–516.
[13] ITU-R. *Propagation Data and Prodeciction Methods for the Planning of Short-Range Outdoor Radiocommunication Systems and Radio Local Area Networks in the Frequency Range 300 MHz to 100 GHz*. Accessed: Dec. 21, 2018. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.1411-9-201706-I!!PDF-E.pdf
[14] LoRa Alliance. *LoRaWAN Regional Parameters v1.1rB*. Accessed: Dec. 28, 2018. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_regional_parameters_v1.1rb_-_final.pdf
[15] C. Goursaud and J.-M. Gorce, "Dedicated networks for IoT: PHY/MAC state of the art andchallenges," *EAI Endorsed Trans. Internet Things, Eur. Alliance Innov.*, 2015. doi: 10.4108/eai.26-10-2015.150597.
[16] R. S. Sutton, A. G. Barto, and F. Bach, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[17] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Mach. learn.*, vol. 8, nos. 3–4, pp. 293–321, May 1992.
[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2010.
[19] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[20] M. K. Johnson and R. M. Liebert, *Statistics: Tool of the Behavioral Sciences*. London, U.K.: Pearson, 1977.
[21] N. Buduma, *Fundamentals of Deep Learning*. Newton, MA, USA: O'Reilly Media, Inc., 2018.
[22] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in loRaWAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.

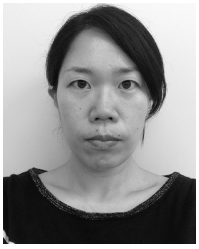


NAOKI AIHARA received the B.E. degree in information and communication engineering from The University of Electro-Communications in 2018, where he is currently pursuing the M.E. degree. His research interests include machine learning and its application to wireless communication.

KOICHI ADACHI received the B.E., M.E., and Ph.D. degrees in engineering from Keio University, Japan, in 2005, 2007, and 2009 respectively. From 2007 to 2010, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow. From May 2010 to May 2016, he was with the Institute for Infocomm Research, A*STAR, in Singapore. He was a Visiting Researcher with the City University of Hong Kong, in April 2009 and the Visiting Research Fellow with the University of Kent from June to Aug 2009. He is currently an Associate Professor with The University of Electro-Communications, Japan. His research interests include cooperative communications and energy efficient communication technologies.

Dr. Adachi was a recipient of the Excellent Editor Award from the IEEE ComSoc MMTC, in 2013. He served as a General Co-Chair for the 10th and 11th IEEE Vehicular Technology Society Asia Pacific Wireless Communications Symposium (APWCS), a Track Co-Chair for Transmission Technologies and Communication Theory of the 78th and 80th IEEE Vehicular Technology Conference, in 2013 and 2014, respectively, and a Symposium Co-Chair for the Communication Theory Symposium of the IEEE GLOBECOM 2018, and a Tutorial Co-Chair for IEEE ICC 2019. He was an Associate Editor of *IET Transactions on Communications*, from 2015 to 2017. He has been an Associate Editor of the IEEE Wireless Communications Letters, since 2016, the IEEE Transaction on Vehicular Technology, from 2016 to 2018, the IEEE Open Journal of Vehicular Technology, since 2019. He was recognized as the Exemplary Reviewer from the IEEE Communications Letters, in 2012 and the IEEE Wireless Communications Letters, in 2012, 2013, 2014, and 2015.

OSAMU TAKYU received the B.E. degree in electrical engineering from the Tokyo University of Science, Chiba, Japan, in 2002, and the M.E. and Ph.D. degrees in open and environmental systems from Keio University, Yokohama, Japan, in 2003 and 2006, respectively. From 2003 to 2007, he was a Research Associate with the Department of Information and Computer Science, Keio University. From 2004 to 2005, he was a Visiting Scholar with the School of Electrical and Information Engineering, University of Sydney. From 2007 to 2011, he was an Assistant Professor with the Department of Electrical Engineering, Tokyo University of Science. From 2011 to 2013, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Shinshu University, where he has been an Associate Professor, since 2013. His current research interests include wireless communication systems and distributed wireless communication technology. He was a recipient of the Young Researcher's Award of IEICE 2010, the 2010 Active Research Award in Radio Communication Systems (RCS) from IEICE Technical Committee on RCS, and the 2018 Best Paper Award in Smart Radio (SR) from IEICE technical committee on SR.



MAI OHTA received the B.E., M.E., and Ph.D. degrees in electrical engineering from The University of Electro-Communications, Tokyo, Japan, in 2008, 2010, and 2013, respectively. Since 2013, she has been an Assistant Professor with the Department of Electronics Engineering and Computer Science, Fukuoka University. Her research interests include cognitive radio, spectrum sensing, LPWAN, and sensor networks. She was a recipient of the Young Researcher's Award from IEICE, in 2013.

TAKEO FUJII received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1997, 1999, and 2002, respectively. From 2000 to 2002, he was a Research Associate with the Department of Information and Computer Science, Keio University. From 2002 to 2006, he was an Assistant Professor with the Department of Electrical and Electronic Engineering, Tokyo University of Agriculture and Technology. From 2006 to 2014, he was an Associate Professor with the Advanced Wireless Communication Research Center, The University of Electro-Communications. He is currently a Professor and the Director of the Advanced Wireless and Communication Research Center, The University of Electro-Communications. His current research interests include cognitive radio and ad-hoc wireless networks.

He is a Fellow of IEICE. He was a recipient of the Best Paper Award in the IEEE VTC 1999-Fall, the 2001 Active Research Award in Radio Communication Systems from the IEICE technical committee of RCS, the 2001 Ericsson Young Scientist Award, the Young Researcher's Award from the IEICE, in 2004, The Young Researcher Study Encouragement Award from IEICE Technical Committee of AN, in 2009, the Best Paper Award in the IEEE CCNC 2013, and the IEICE Communication Society Best Paper Award, in 2016.