applied sciences

MDPI

Article

A Reinforcement-Learning-Based Distributed Resource Selection Algorithm for Massive IoT

Jing Ma [1,*,†], So Hasegawa [1,†] , Song-Ju Kim [2] and Mikio Hasegawa [1,†]

[1] Department of Electrical Engineering, Tokyo University of Science, Tokyo 125-8585, Japan
[2] Graduate School of Media and Governance, Keio University, Fujisawa, Kanagawa 252-0882, Japan
* Correspondence: jing.ma.jb@gmail.com
† Current address: Department of Electrical Engineering, Graduate School of Engineering, Katsushika Campus, 6-3-1 Niijyuku , Katsushika-ku, Japan.

Received: 27 June 2019; Accepted: 2 September 2019; Published: 6 September 2019

check for updates

Abstract: Massive IoT including the large number of resource-constrained IoT devices has gained great attention. IoT devices generate enormous traffic, which causes network congestion. To manage network congestion, multi-channel-based algorithms are proposed. However, most of the existing multi-channel algorithms require strict synchronization, an extra overhead for negotiating channel assignment, which poses significant challenges to resource-constrained IoT devices. In this paper, a distributed channel selection algorithm utilizing the tug-of-war (TOW) dynamics is proposed for improving successful frame delivery of the whole network by letting IoT devices always select suitable channels for communication adaptively. The proposed TOW dynamics-based channel selection algorithm has a simple reinforcement learning procedure that only needs to receive the acknowledgment (ACK) frame for the learning procedure, while simply requiring minimal memory and computation capability. Thus, the proposed TOW dynamics-based algorithm can run on resource-constrained IoT devices. We prototype the proposed algorithm on an extremely resource-constrained single-board computer, which hereafter is called the cognitive-IoT prototype. Moreover, the cognitive-IoT prototype is densely deployed in a frequently-changing radio environment for evaluation experiments. The evaluation results show that the cognitive-IoT prototype accurately and adaptively makes decisions to select the suitable channel when the real environment regularly varies. Accordingly, the successful frame ratio of the network is improved.

Keywords: reinforcement learning; multi-armed bandit; IoT; distributed channel selection

1. Introduction

Massive Internet of Things (IoT) has gained great research attention. Its objective is to connect a wide range of devices to share data and information with each other. Massive IoT targets various applications such as smart cities, environmental monitoring, and health-care monitoring systems. Computing Technology Industry Association (CompTIA) has predicted that the number of connected devices will reach over 50 billion units by 2020 and 125 billion by 2030 [1]. However, the majority of such connected devices have low processing power, limited storage capacity, and energy constraints.

The network congestion becomes an issue as the sheer number of connected devices increases over a network. IoT devices generate enormous traffic on the communication path, which creates network congestion. To manage network congestion, a multi-channel technology is proposed to be applied to the wireless communication network, which can benefit from the parallel transmission and reduced interference. Numerous multi-channel-based algorithms [2,3] are proposed to dynamically select channels for communication for improving the IoT network performance. All these related works are proposed based on time-slotted channel hopping (TSCH), which is a mechanism to improve the reliability of an IoT network and included in the IEEE 802.15.4e standard [4]. The fundamental principle of TSCH is the scheduling of the time slots and channels for each communication in an IoT network, which can simultaneously decrease the frequency of communications and improve the network performance.

However, most of the existing multi-channel algorithms require strict synchronization, an extra overhead for negotiating channel assignment, which poses significant challenges to resource-constrained IoT devices. In addition, the current available IoT devices, especially resource-constrained IoT devices, have only one simple half-duplex radio transceiver, which does not satisfy the strong assumptions of operating on multiple channels simultaneously in most of existing works.

In contrast, to implement self-decision and self-learning in a wireless network system, recently, numerous reinforcement-learning-based approaches have been proposed to improve the wireless communication systems. In [5], a reinforcement learning solution and simulation analysis were given for heterogeneous cellular networks to select multiple radio access technologies (RAT). Channel assignment schemes were proposed for cognitive radio networks (CRNs) that address the tradeoff of rate maximization and network connectivity in [6,7]. The problem of assigning channels has been modeled as a multi-armed bandit (MAB) [8] problem, and accordingly, the MAB algorithm was proposed to explore and exploit suitable channel assignment.

The work in [9–11] proposed an efficient, yet simple MAB algorithm, called the tug-of-war (TOW) dynamics, which could make decisions accurately and rapidly following a change in the environment. The work in [12,13] proposed to apply the TOW dynamics to a cognitive radio system and WLAN. In the above-mentioned related work, the experiments showed network performance improvement on the application of the TOW dynamics.

In this paper, a reinforcement-learning-based channel selection algorithm utilizing the TOW dynamics [9–11] is proposed. In this proposal, a simple learning procedure for updating the reward estimates only needs to check whether the ACK frame is received or not. Meanwhile, minimal memory and computation capability for the learning procedure are required. Thus, the proposal can run on resource-constrained IoT devices. The proposal only focuses on the medium access control (MAC) layer procedure modification. We prototype the proposed algorithm on a resource-constrained single-board computer (SBC), which is hereafter called the cognitive-IoT prototype. Moreover, evaluation experiments deploying the cognitive-IoT prototype with high density in a frequently-changing radio environment are conducted. The evaluation results show that the cognitive-IoT prototype accurately and adaptively makes decisions to select channels respecting fairness among IoT devices when the real environment regularly varies.

2. IoT System

2.1. IEEE 802.15.4-Oriented Wireless Technology for IoT

The IEEE 802.15.4 standard [4], characterized by low power, low cost, high reliability, and simplicity in the industrial, scientific, and medical (ISM) frequency band, is one of the key enabling technologies for IoT deployment. Therefore, the IEEE 802.15.4-oriented wireless devices have been widely deployed in many applications such as smart homes, environmental monitoring, and industrial automation.

The IEEE 802.15.4g amendment [14] defines new PHY in the 920-MHz band for smart utility application and focuses on range. Three alternative PHYs, i.e., frequency shift keying (FSK), offset quadrature phase shift keying (O-QPSK), and orthogonal frequency division multiplexing (OFDM) are supported. FSK and O-QPSK are conventional well-known modulation techniques, while OFDM is commonly used in advanced systems. Any IEEE 802.15.4g-compliant chip must implement a PHY with two-FSK modulation and a 50-kbps data rate. All other PHYs are optional. Each PHY allows further parametrization with data rates ranging from 6.25 kbps–800 kbps. In this work, the proposal only focuses on the MAC layer procedure, which is introduced as below.

Depending on the massive IoT requirements, a PAN consists of one PAN coordinator or a gateway and multiple sensors and transmits the data to the coordinator or gateway with a simple star topology. In addition, the carrier sense multiple access algorithms with collision avoidance (CSMA/CA) algorithm is utilized as the radio channel access method for IoT devices. To satisfy the requirements of massive IoT, typically, a non-beacon-enabled mode, employing a simple unslotted CSMA/CA and direct transmission, is used to relieve the strict synchronization and minimize unnecessary protocol overheads. Three variables for attempting transmission, i.e., number of back-offs (NB), contention window (CW), and back-off exponent (BE), are maintained for each device. NB is the number of times required to back-off while attempting the current transmission based on the CSMA/CA algorithm. CW defines the number of back-off periods that need to be clear of the channel activity before transmitting. BE is the back-off exponent to determine the number of slots that a device has to wait before accessing a channel.

Figure 1 describes an unslotted CSMA/CA algorithm. Initially, the NB, CW, and BE are initialized as 0, 2, and $macMinBE$, respectively. Then, the device delays for a random back-off duration in the range from 0–$2^{BE} - 1$. Subsequently, the PHY layer detects whether the channel is idle or not by performing a clear channel assessment (CCA). If the channel is assessed as idle, the MAC layer proceeds with the remaining CSMA/CA algorithm steps, including the frame transmission and acknowledgment (ACK) frame reception. If the ACK is received correctly, it terminates with a transmission success status, otherwise it terminates with a transmission failure status. If the channel is assessed to be busy, the MAC layer increments both the NB and BE by one, ensuring that the $BE \leq aMaxBE$, and the CW is set as two. If $NB \leq macMaxCSMABackoffs$, the CSMA/CA algorithm starts the back-off again. Otherwise, it terminates with a failure status.

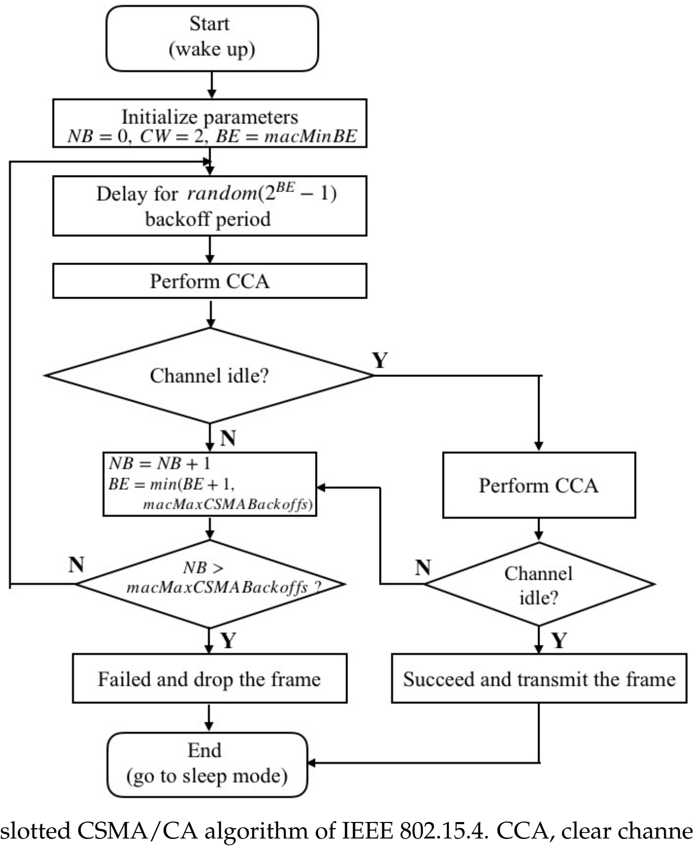

Figure 1. Unslotted CSMA/CA algorithm of IEEE 802.15.4. CCA, clear channel assessment.

2.2. System Model

In this work, the studied IoT system model is considered to be maximally reasonably simple for the target applications of massive IoT, such as a monitoring application. First, an IoT network, where IoT devices that can only select one of multiple available channels to deliver the frame each time, was established. Extremely simple applications such as a switch or a sensor were mainly considered for each resource-constrained IoT device. Therefore, it was not intended for exchanging numerous frames with much resource usage and large memory capacity. Consequently, these IoT devices were resource constrained, using a battery, equipping one simple half-duplex transceiver, and turning off the radio periodically to save energy. Thus, it was practically difficult to run a complex algorithm on it.

Moreover, generally, IoT network topologies are star- or tree-based, with the data collected by groups of sensors and sent to a collector or border router. In this work, the IoT devices are assumed to be associated with a gateway with a star topology, as shown in Figure 2, because the star topology is simple enough so that power consumption could be reasonably low for resource-constrained IoT devices. To minimize power consumption, resource-constrained IoT devices assumed in this work do not conduct any complicated process such as channel measuring, data frame receiving, or forwarding.

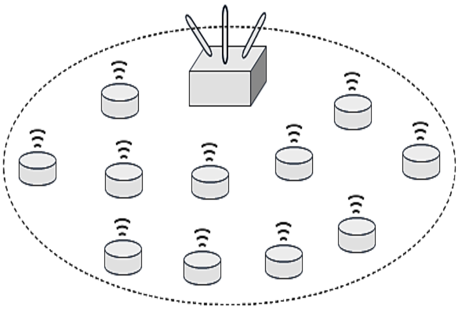

Figure 2. System model.

In addition, each IoT device accesses the channel for transmission following the procedure of IEEE 802.15.4 non-beacon-enabled mode, in which synchronization is not necessary. Each time an IoT device wakes up when it has to transmit a data frame, it selects a channel for the data frame transmission and performs the CSMA/CA algorithm (see Figure 1), as discussed in Section 2.1, to send the data frame. Regardless of the success of the current frame transmission, the IoT device goes to the sleep mode for a predetermined "sleep interval" duration to save power.

3. Problem Formulation

3.1. Multi-Armed Bandit Problem

MAB [8] is a statistical model that balances exploration and exploitation to solve recurring decision problems. A common real-world comparison for this problem is a gambler facing a collection of slot machines ("bandits") at a casino, with each machine having an "arm" to pull. Each machine has an unknown distribution and expected payout, and the goal is to select arms that maximize the winnings by a sequence of lever pulls. After each attempt, the gambler must decide which slot machine to play given the current knowledge about their payouts.

Formally, an MAB is defined with K arms with unknown reward probabilities $(p_1, p_2, ..., p_K)$, which is the fundamental challenge of solving the MAB problem. At each round t, the player plays arm $k \in 1, ..., K$ and obtains a reward. Therefore, the player has to play machines strategically to both maximize the reward and discover information about the arm rewards, denoted as X_k. The target is to maximize the accumulated rewards, $\mathbb{E}\left\{\sum_{t=1}^{\infty} X_t\right\}$.

3.2. Channel Selection Problem as an MAB Problem

According to the system model introduced in Section 2.2, the objective of the distributed channel selection problem for IoT devices is to maximize the total frames successfully transmitted via an optimal channel selection strategy. We assumed distributed channel selection processes for multiple IoT devices with no prior coordination. Each IoT device can select a set $K = \{1, ..., k\}$ of available channels. Here, k refers to the index of the channel. For simplicity, we assumed that each IoT device can access one channel at a time. The distributed channel selection problem of IoT device belongs to the class of MAB.

The challenge in distributed channel selection problem is similar to that in the MAB problem: an IoT device (player) has K channels (slot machines), the k^{th} of which has successful frame transmission possibility $p_k, k \in K$ (reward probability). The IoT device does not know the values of the p_ks and must sequentially select a channel to send the data frame (select a machine to play). The target is to maximize the total successful frame transmission (overall gain) for a total of T transmissions (plays). The distributed channel selection problem and MAB problem share the same concept, which is summarized in Table 1.

Table 1. Channel selection problem as the multi-armed bandit (MAB) problem.

MAB Problem	Distributed Channel Selection Problem
Player	IoT device
Slot machine	Wireless channel
Reward: coin	Reward: number of transmission successful frames over the selected channel
Objective: maximize the total number of coins	Objective: maximize the total number of successful frame transmissions

An IoT device selects channel $k^*(t) \in \mathcal{K}$ at current time t. If the IoT device sends the data frame over a selected channel k^* successfully, the selected channel, k^*, receives a reward. We assume that the IoT devices follow a generalized version of the unslotted CSMA/CA algorithm introduced in Section 2.2 For more details, if channel $k^*(t)$ is selected, an IoT device will wait the time specified by the generated random number. At the end of the waiting period, the IoT device senses the selected channel again, and if it is found to be busy, it will generate another random number following the CSMA/CA algorithm. It will then detect the selected channel, $k^*(t)$, again until reaching the predetermined maximum challenge time. If the selected channel, $k^*(t)$, is still found busy when reaching the maximum retry time, the IoT device will discard the data frame and recognize the current data frame transmission as failed.

However, if the selected channel, $k^*(t)$, is assessed as idle, the data frame will be transmitted by the IoT device. Moreover, if the ACK frame is received successfully from the destination of the data frame transmitted previously, then the IoT device will recognize the transmitted data frame over the selected channel, $k^*(t)$, as a success. If the ACK frame is not received successfully from the destination of the data frame transmitted previously, then the IoT device will recognize the transmitted data frame over the selected channel, $k^*(t)$, as failed. Regardless of the success of the current data transmission, the IoT device will enter sleep and possibly select a different channel to access when it wakes up the next time.

The challenge stems from the uncertainty in the successful frame transmission probability, $p_k(t)$, imposing a trade-off between the exploration learning $p_k(t)$, and exploitation by selecting the channel with the highest estimated successful frame transmission probability, $p_k(t)$, based on the currently available information. An IoT device employs a strategy that will select channel $k \in K$ to access at current time t according to any possible causal information pattern obtained from the previous $t - 1$ observations. This is done to maximize the accumulated reward, $X_k(t)$, i.e., the total successful frame transmission.

4. TOW Dynamics-Based Strategy

Many algorithms [15–18] have been proposed to solve the MAB problem. In this paper, the strategy used to solve the distributed channel selection problem is referred to the tug-of-war (TOW) dynamics [9–12]. Despite the simplicity, the high efficiency of TOW dynamics has been analytically validated in making a series of decisions for maximizing the total sum of stochastically-obtained rewards in an environment where the reward probability frequently changes [9–12]. The main methodology of [9–12] is summarized below. The essential element of the TOW dynamics is the consideration of a volume-conserving physical object, e.g., the incompressible liquid (blue) is assumed in a branched cylinder shown in Figure 3, which implies a non-local correlation between the terminal parts. Specifically, the volume increase in one part is immediately compensated by the volume decrease in the other part(s). Here, the n-machine MAB is considered. For each machine (branch) k at time t, let $X_k(t)$ correspond to the displacement of machine k. Thus, the reward estimates of each machine can be obtained by Equation (1),

$$Q_k(t) = N_k(t) - (1+\omega)L_k(t) = \;\; Q_k(t-1) + \Delta\, Q_k(t) \tag{1}$$

where $k \in (1, ..., n)$. Here, N_k denotes the accumulated counts of selections of machine k, while L_k denotes non-rewarded counts of machine k until time t. $\Delta\, Q_k(t)$ is given by Equation (2).

$$\Delta Q_k(t) = \begin{cases} +1 & \text{if rewarded} \\ -\omega & \text{if non-rewarded} \end{cases} \tag{2}$$

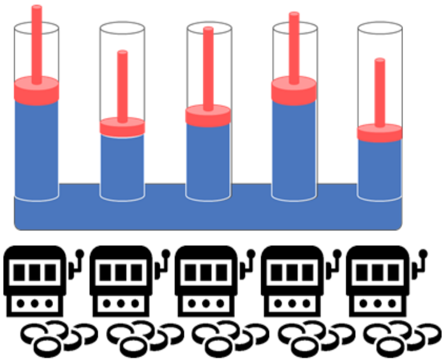

Figure 3. tug-of-war (TOW) dynamics for n-machine MAB.

Accordingly, based on the conservation laws in TOW dynamics, the displacement of machine k can be estimated by Equation (3),

$$X_k(t+1) = Q_k(t) - \frac{1}{n-1} \sum_{k' \neq k} Q_{k'}(t) + osc_k(t) \tag{3}$$

where n is the total number of arms. The incompressible liquid oscillates autonomously according to Equation (4) [12]. There are many possibilities of adding the oscillations. In [11], the influence of *osc* on the efficiency of decision making as studied, which is outside of the scope of this paper. For the sake of simplicity, we used $A = 0.5$ for all machines in this paper.

$$osc_k(t) = A cos(2\pi t/n + 2(k-1)\pi/n) \tag{4}$$

The estimated reward probability, denoted as $p_k(t)$, can be obtained by Equation (5),

$$p_k(t) = \frac{R_k(t)}{N_k(t)}. \tag{5}$$

where $R_k(t) = N_k(t) - L_k(t)$ counts the number of times of getting rewards when machine k is played N_k times until time t. Consequently, the TOW dynamics-based strategy evolves according to a particular simple rule: if machine k is played at each time t, $+1$ or $-\omega$ is added to $X_k(t+1)$ when rewarded and non-rewarded, respectively. To explore the appropriate weight parameter of ω, the expected value of reward estimates of machine k can be obtained by Equation (6).

$$\mathbb{E}[Q_k(t)] = \{p_k - \omega(1-p_k)\}N_k \tag{6}$$

Here, the highest and the second highest estimated reward probability among n machines can be obtained given by Equation (5), denoted as p_{1st} and p_{2nd}, respectively. Accordingly, the highest and the second highest expected value of reward estimates among n machines can be obtained by Equation (6). To achieve always selecting the machine with the highest reward estimates, the following forms could be expressed,

$$\begin{aligned} p_{1st} - \omega(1-p_{1st}) &> 0 \\ p_{2nd} - \omega(1-p_{2nd}) &< 0 \end{aligned} \tag{7}$$

and these expressions can be rearranged into the form:

$$p_{1st} < \frac{\omega}{1+\omega} < p_{2nd} \tag{8}$$

In other words, the weight parameter ω should satisfy the above condition Inequality (8) so that the selection correctly represents the largest reward estimates. One way of ensuring Inequality (8) is to take an ω that satisfies Equation (9).

$$\frac{\omega}{1+\omega} = \frac{p_{1st} + p_{2nd}}{2}. \tag{9}$$

From Equation (9), we obtain the appropriate value of weight parameter ω given by Equation (10) to use the TOW dynamics-based strategy for selecting the correct machine that maximizes the reward.

$$\omega = \frac{p_{1st} + p_{2nd}}{2 - p_{1st} - p_{2nd}}. \tag{10}$$

It is implied from the above description that the TOW algorithm has an equivalent learning rule for a system that can update the reward estimates simultaneously, which would be able to make decision with high efficiency for a variety of real-world applications. In [12,13], the TOW dynamics was proposed to be utilized in wireless cognitive radio and WLAN systems respectively to improve the multi-user channel allocations of the cognitive radio efficiently.

In this paper, the TOW dynamics-based strategy is proposed to solve the distributed channel selection problem in massive IoT, which is modeled as the MAB problem (see Section 3.2). The objective is to maximize the total successful frame transmissions. The TOW-dynamics-based strategy is applied to explore the appropriate channel selection in the proposal by simply checking the ACK frame reception. The learning rule of the proposal is based on Equation (1).

Meanwhile, the estimated reward probability $p_k(t)$ (see Equation (5)) is actually obtained by Equation (11) according to the massive IoT system procedure in the proposal.

$$p_k(t) = \frac{\text{No. of ACK received over channel } k \text{ until time } t}{\text{No. of data frame sent over channel } k \text{ until time } t} \tag{11}$$

Then, the appropriate ω can be explored as given by Equations (10) and (11).

In addition, because the wireless channels are frequently changing, a parameter, denoted as the forgetting ratio α, is used for controlling how much the past experiences influence. Then, $Q_k(t)$ is proposed to be rewritten as Equation (12).

$$Q_k(t) = \ \alpha Q_k(t-1) + \Delta\ Q_k(t) \quad (0 < \alpha \leq 1) \tag{12}$$

Above, the closer α is to zero, the fewer estimated $Q_k(t)$ are influenced by the past experiences. In contrast, the closer α is to one, the more estimated $Q_k(t)$ are influenced by the past experiences. Accordingly, $X_k(t)$ of each available channel can be determined by Equation (3). Then, the IoT device will select the channel, $k^* = \arg\max_{k \in K} X_k(t+1)$, once it wakes up the next time.

5. Proposal: TOW Dynamics-Based Channel Selection Algorithm and Its IoT Prototype Implementation

In this section, the proposed TOW dynamics-based channel selection algorithm and its prototype implementation are described. The proposed algorithm works extremely simply. Accordingly, only minimal memory and computation capability are necessary to implement the algorithm on a resource-constrained IoT device. The IoT device conducts the proposed channel selection algorithm to update the accumulated reward estimates, $X_k(t+1)$, of the all available channels simultaneously following the learning process described in Section 4.

Based on the system model and problem formulation introduced in Sections 2.2 and 3.2 respectively, the IoT device initially wakes up and randomly selects a channel from the available K channels. As mentioned in Section 3.1, if the IoT device sends the data frame over selected channel k^* and receives the ACK frame from the destination, then the IoT device will recognize the data transmission as a success over the selected channel, k^*, which implies that it is rewarded, and ΔQ_{k^*} given by Equation (2) is set as $+1$. Otherwise, if the IoT device does not receive the ACK frame, it will recognize the data transmission over the selected channel, k^*, as failed, which implies that it is unrewarded, and ΔQ_{k^*} is set as $-\omega(t)$.

Figure 4 shows the flowchart of the proposed procedure for each IoT device. Essentially, the proposed procedure follows the same procedure as that of the non-beacon-enabled mode of IEEE 802.15.4. Therefore, the IoT device periodically goes to the sleep mode to save power after sending a data frame. Each IoT device initializes $Q_k(0)$, $R_k(0)$, and $N_k(0)$ at $t = 0$. Once the IoT device wakes up at $t = 0$, it sends the data frames over the selected channel $k^*(0) = \arg\max_{k \in K} X_k(0)$. Because the $X_k(0)$ is randomly set to the same value for all available channels initially, the IoT device actually randomly selects a channel to send data frames at $t = 0$. Meanwhile, the count of transmissions over channel $k^*(0)$, denoted as $N_{k^*(0)}$, is incremented by one.

If the IoT device receives the ACK frame successfully, the channel $k^*(0)$ is rewarded, and $\Delta Q_{k^*}(0) = +1$ is added to $Q_{k^*}(0)$. Meanwhile, the counts of rewards $R_{k^*}(0)$ is incremented by one accordingly. Otherwise, the selected channel $k^*(0)$ is unrewarded, and $\Delta Q_{k^*}(0) = -\omega(0)$ is added to $Q_{k^*}(0)$ for updating. Regarding those unselected channels $k(k \neq k^*)$ at $t = 0$, $\Delta Q_{k(k \neq k^*(0))}(0) = 0$. The reward estimates of the unselected channels are given by $Q_{k(k \neq k^*(0))}(1) = Q_k(0)$ accordingly.

Then, $X_k(1)$ of each available channel is updated by Equation (3) for the next time $t = 1$. Meanwhile, the reward probability estimate of the selected channel k^*, i.e., $p_{k^*(1)}$, is updated for the next time $t = 1$ according to Equation (11). In addition, the appropriate value of weight parameter $\omega(1)$ is also updated adaptively based on Equation (10) for the next time $t = 1$ Then, the IoT device goes to sleep mode again and wakes up at $t++$ $(t = 1\ here)$ to send a data frame over the selected channel with the highest $X_k(1)$, which has been calculated at $t = 0$.

We implemented the proposed channel selection algorithm on an SBC shown in Figure 5. As described above, the cognitive-IoT prototype only needs to receive the ACK frame to explore the channel selection strategy. Concurrently, the proposed TOW dynamics-based learning process only requires adding one or subtracting ω to update the reward estimate. Therefore, to implement the proposal, only addition and subtraction are necessary based on the procedure of the proposal. Thus, minimal memory and computation capability are enough for prototyping the proposed reinforcement-learning-based channel selection algorithm. More details of the processes of the algorithm running on the cognitive-IoT prototype are summarized in Algorithm 1.

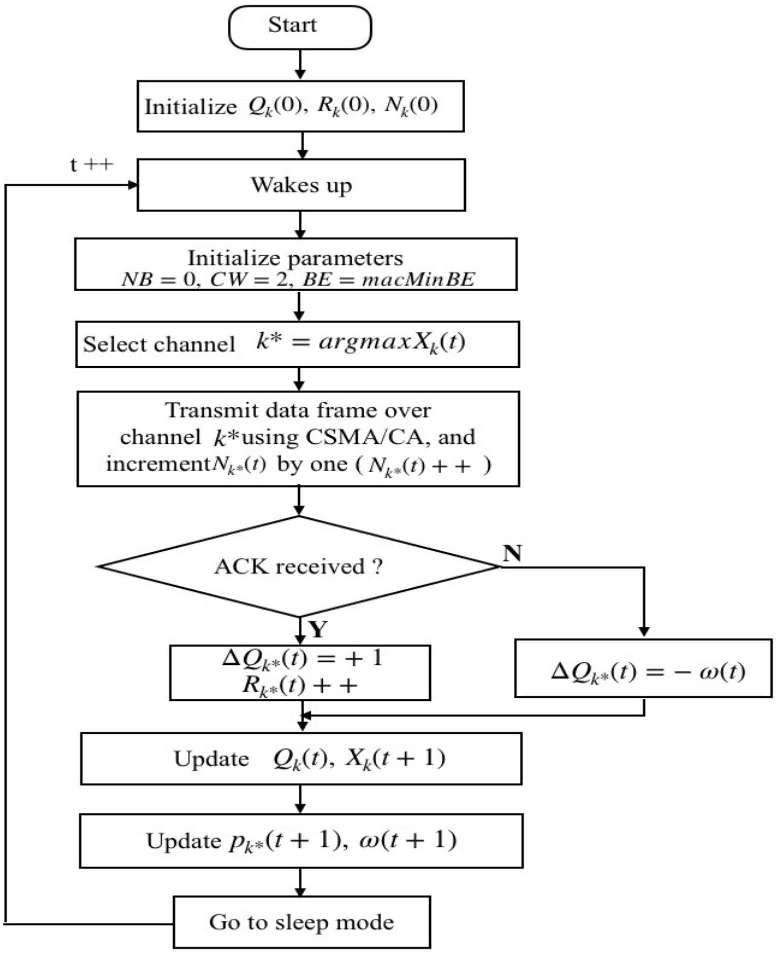
Start
Initialize $Q_k(0)$, $R_k(0)$, $N_k(0)$
t ++
Wakes up
Initialize parameters
$NB = 0$, $CW = 2$, $BE = macMinBE$
Select channel $k^* = argmaxX_k(t)$
Transmit data frame over channel k^*using CSMA/CA, and increment$N_{k^*}(t)$ by one ($N_{k^*}(t)++$)
ACK received ?
N
Y
$\Delta Q_{k^*}(t) = +1$
$R_{k^*}(t)++$
$\Delta Q_{k^*}(t) = -\omega(t)$
Update $Q_k(t)$, $X_k(t+1)$
Update $p_{k^*}(t+1)$, $\omega(t+1)$
Go to sleep mode

Figure 4. Flowchart of the proposed scheme.

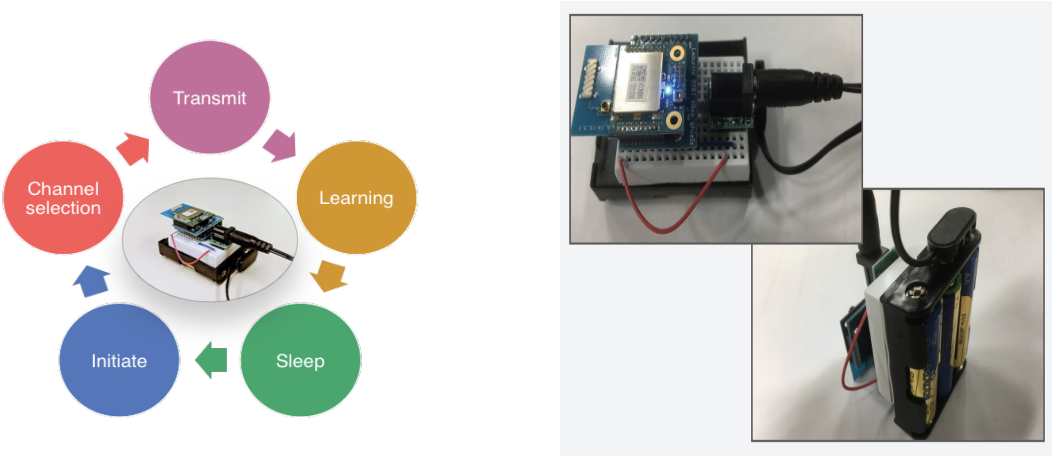

Figure 5. Cognitive-IoT prototype.

Algorithm 1 TOW dynamics-based channel selection algorithm for the cognitive-IoT prototype $m \in M$.

```
Initialize $Q_k(0),\ R_k(0),\ N_k(0)$.
while wake time $t = 1, 2, 3...$ is not expired do
    Initiate channel access parameters, i.e., NB, CW, BE
    Select channel $k^* = argmax X_k(t) (\forall k \in K)$
    while $NB < macMaxCSMABackoffs$ do
        Delay for random $(2^{BE} - 1)$
        Perform clear channel assessment (CCA)
        if Channel is assessed as idle then
            Transmit the data frame
            $N_{k^*}(t) + +$
            if the data frame is transmitted, and the corresponding ACK frame is received then
                Data transmission succeeded
                Set $\Delta Q_{k^*}(t) = +1$
                Increment $R_{k^*}(t)$ by one $(R_{k^*}(t) + +)$
            else
                Data transmission failed
                Set $\Delta Q_{k^*}(t) = -\omega(t)$
            end if
            Update $Q_{k(t)}$ given by Equation (12)
            Update $X_k(t + 1)$ given by Equation (3)
            Update $p_{k^*}(t + 1)$ given by Equation (11)
            Update $\omega(t + 1)$ given by Equation (10)
        else if Channel is assessed as busy then
            Update $NB = NB + 1$ and $BE = min(BE + 1, macMaxBE)$
            Continue
        end if
    end while
    Enter sleep mode for 1000 ms
    $t = t + 1$
end while
```

The cognitive-IoT prototype was developed on a well-used SBC, Lazurite 920 MHz [19]. Lazurite is equipped with the IEEE 802.15.4g radio-compatible transceiver supporting a 50-kbps data rate in the 920-MHz band. An ultra-low power 16-bit microcontroller (ML620Q504H) and 64 KB ROM are used in Lazurite. In addition, Lazurite's operating voltage range is 1.8 V∼5.5 V, while the operating current range is 7 µA∼mA The photographs of the cognitive-IoT prototype are shown in Figure 5.

6. Performance Evaluation

To validate that the cognitive-IoT prototypes make the decisions to select the suitable channels to transmit adaptively, a series of experiments were conducted. A photograph of the experiment setting is shown in Figure 6. The cognitive-IoT prototypes (see Figure 7) were deployed in a 1 m × 1 m area with high density, where they communicated with three gateways (see Figure 8) operating on different channels with a star topology, following the proposed procedure introduced in Section 5. The deployment and topology were static throughout the experiments.

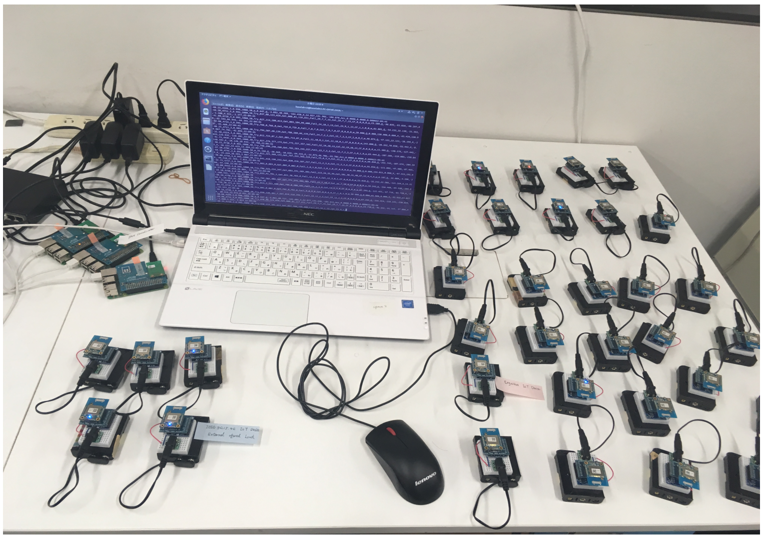

Figure 6. Photograph of the experiment setting.

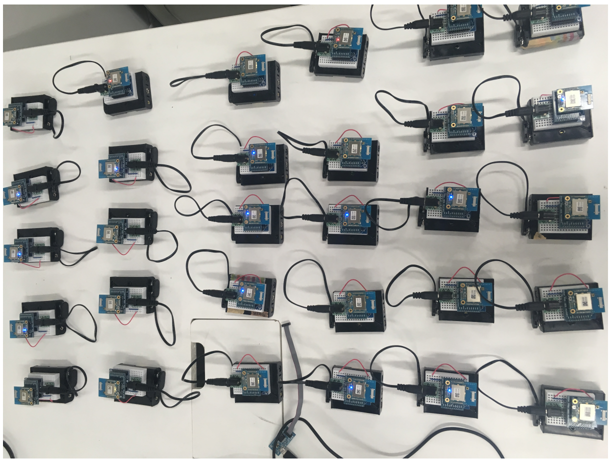

Figure 7. Cognitive-IoT prototypes.

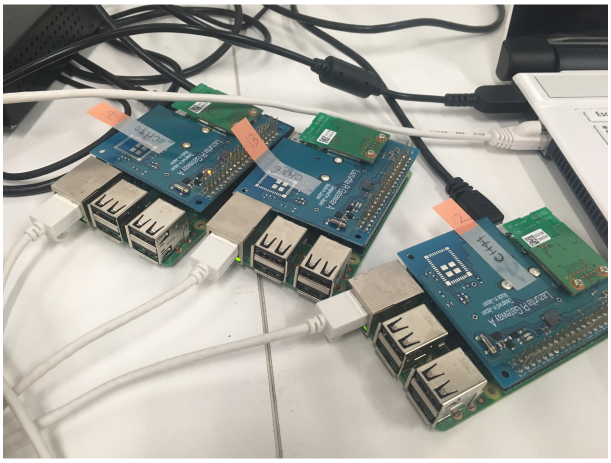

Figure 8. Gateways.

In addition, the bandwidth of all three available channels was 200 kHz. Each IoT node had a 1000-ms sleep interval. α was chosen as 0.995 in the experiments. Table 2 summarizes the experiment parameters.

Table 2. Experiment Parameters.

Parameters	Worth
Available channels	924.6 MHz (CH44), 925.8 MHz (CH50), 927.0 MHz (CH56)
Number of cognitive-IoT prototypes	30
Data rate (kbps)	50
Transmission power (mw)	20
Sleep mode interval of the cognitive-IoT prototype (ms)	1000
α	0.995

Furthermore, to validate that the cognitive-IoT prototypes select suitable channels adaptively, the externally-offered loads that were generated by the IEEE 802.15.4e IoT devices were added to evaluate the effectiveness of the cognitive-IoT prototypes in four different test scenarios, which are summarized in Table 3. The sleeps internal of all the IEEE 802.15.4e IoT devices was set as 100 ms.

Table 3. Externally-offered load added in the experiment environment.

Test Scenario	Details
$Ld = (0, 0, 0)$	No externally-offered load added
$Ld = (0, 2, 3)$	Two IEEE 802.15.4e IoT devices operating on CH50 are added, and three IEEE 802.15.4e IoT devices operating on CH56 are added
$Ld = (0, 1, 4)$	One IEEE 802.15.4e IoT device operating on CH50 is added, Four IEEE 802.15.4e IoT devices operating on CH56 are added
$Ld = (0, 0, 5)$	Five IEEE 802.15.4e IoT devices operating on CH56 are added

First, to evaluate that the cognitive-IoT prototypes can improve the successful frame delivery of the network by adaptively selecting the suitable channel to send the frame, the Frame Successful Ratio (FSR) (see Equation (13)) was compared with that of the evenly assigned (EA) IoT devices. Here, the EA IoT devices represent the IEEE 802.15.4e IoT devices on which we implemented the standard procedure of IEEE 802.15.4e on the same SBC with the cognitive-IoT prototype shown in Figure 5. These EA IoT devices were set up to operate statically on the assigned channels where devices were EA on each available channel. Because the EA IoT devices operated based on the standard procedure of IEEE 802.15.4e, the EA IoT devices were not be able to change their channel once their operating channel was pre-determined. Each EA IoT device periodically went to sleep to save energy, which is the same as the cognitive-IoT prototype. The sleep intervals of all the EA IoT devices were set to the same as those of the cognitive-IoT prototypes, i.e., 1000 ms.

$$FSR = \frac{\sum_{j=1}^{M} \sum_{i=1}^{K} R_i^j}{\sum_{j=1}^{M} \sum_{i=1}^{K} N_i^j} \tag{13}$$

where M is the number of devices and, accordingly, j is the index of the device. Meanwhile, K is the number of available channels, and i is the index of the available channel. The FSR results are shown in Figure 9. In the $Ld = (0, 0, 0)$ test scenario, the FSR of the cognitive-IoT prototype was practically the same as that of the EA IoT devices. As for the other test scenarios with an added externally-offered load, i.e., $Ld(0, 2, 3)$, $Ld(0, 1, 4)$, and $Ld(0, 0, 5)$, the FSR results of the cognitive-IoT prototypes were typically higher than those of the EA IoT devices. This was because the cognitive-IoT prototypes generally adaptively selected appropriate channels to send the data frames successfully, whereas the EA IoT devices used the assigned channels, which became congested and failed to send the data frames. It can be stated that by employing the proposed TOW-dynamics-based algorithm to select suitable channels adaptively, the cognitive-IoT prototype improved the FSR of the network.

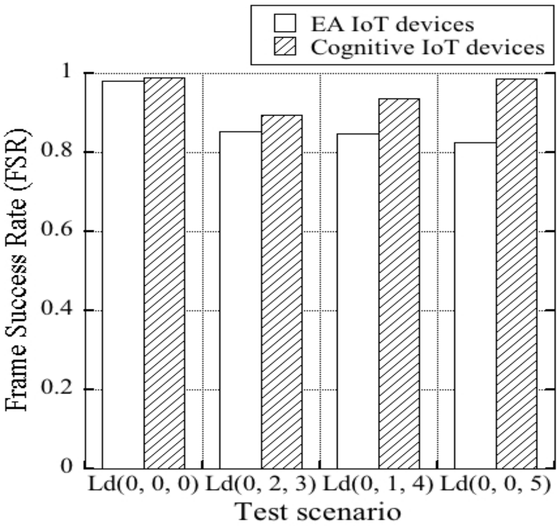

Figure 9. Frame Successful Ratio (FSR) results.

Second, because it was important for the IoT devices to send the data frames unbiasedly, the well-used Jain's fairness index (FI) (see Equation (14)) for the data transmission of the IoT devices was measured.

$$FI(p^1, p^2, ..., p^M) = \frac{(\sum_{j=1}^{M} p^j)^2}{M \sum_{j=1}^{M} (p^j)^2} \tag{14}$$

where M is the number of devices and, accordingly, j is the index of device, while $0 \leq FI \leq 1$, A larger value represents a more reasonable frame transmission achieved by the IoT devices. Figure 10 shows the results of the FIs of the cognitive-IoT prototypes and EA IoT devices.

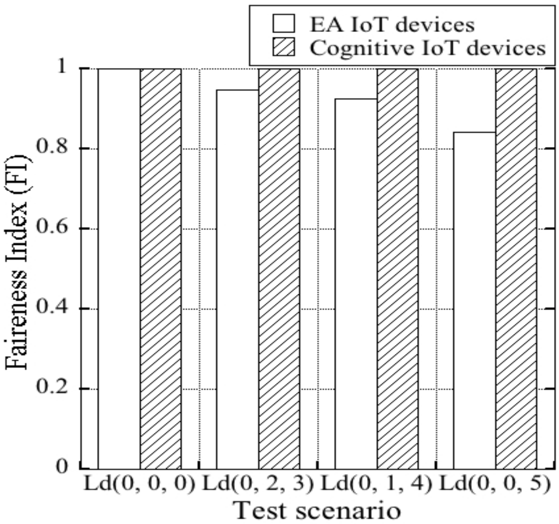

Figure 10. FI results.

As shown in Figure 10, the FI values of both the cognitive-IoT prototypes and EA IoT devices were nearly one when all the available channels were equivalently "idle", i.e., *Ld*: (0, 0, 0). However, for the other test scenarios, i.e., *Ld*: (0, 2, 3), *Ld*: (0, 1, 4), and *Ld*: (0, 0, 5), the FI values of the EA IoT devices decreased because some of the EA IoT devices failed to send the data frames owing to channel congestion, whereas some of them can do this successfully This led to unfairness in the EA IoT devices. By contrast, in all the test scenarios, the FI values of the cognitive-IoT prototypes were typically kept nearly one.

Furthermore, for validating that the cognitive-IoT prototypes adaptively selected the channels, in the experiments, the number of successful frame transmissions over each available channel was observed. We initiated the experiment with test scenario *Ld*: (0, 0, 0) and then changed the test scenarios in the middle of the experiment, i.e., testing *Ld*: (0, 0, 5) for 10 min, *Ld*: (0, 1, 4) for 15 min, *Ld*: (0, 2, 3) for 15 min, and *Ld*: (0, 0, 0) for 15 min subsequently (see Figure 11). The results were tracked and are shown in Figure 11. It can be observed from Figure 11 that the number of successful frame transmissions over the CH56 decreased to almost zero immediately after the test scenario changed to *Ld*: (0, 0, 5) when an externally-offered load was added to CH56. In addition, it is observed from Figure 11 that the cognitive-IoT prototypes started to select CH56 again when the externally-offered load became lighter, i.e., *Ld*: (0, 1, 4) and *Ld*: (0, 2, 3), than before, i.e., *Ld*: (0, 0, 5). It can be stated that the cognitive-IoT prototype achieved selecting the optimal channel accurately and adaptively.

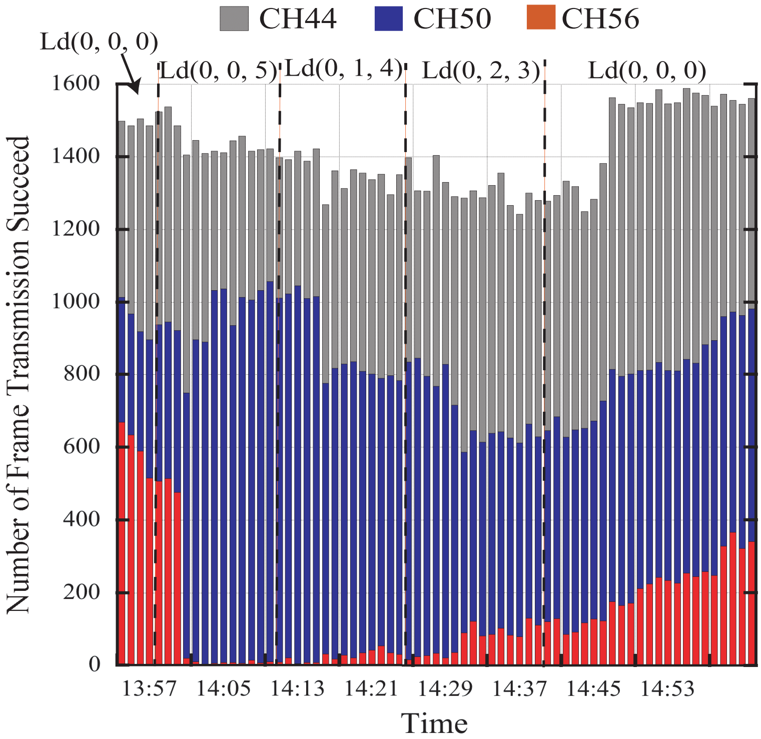

Figure 11. Number of frame transmissions over each available channel.

7. Conclusions

In this paper, a reinforcement-learning-based channel selection algorithm applying TOW dynamics was proposed. The proposal had a simple learning processes that only needed to check whether the ACK frame was received to update the reward estimates. Concurrently, simple learning processes only needed minimal computation capability and memory. Thus, the proposal was able to run on resource-constrained IoT devices. We prototyped the proposal on a resource-constrained SBC. In addition, evaluation experiments densely deploying the cognitive-IoT prototypes in a often changing wireless environment were conducted. The evaluation results showed that the cognitive-IoT prototypes accurately and adaptively made decisions to select channels respecting fairness among IoT devices when the real environment regularly varied. In future work, the proposed reinforcement-learning-based channel selection scheme can be used to select the channel considering more aspects such as channel quality. Moreover, to evaluate the utilization of the developed cognitive-IoT prototype to massive IoT application such as the monitoring system, field experiments that deployed cognitive-IoT prototypes in an architecture monitoring IoT system [20] will be conducted in further research.

Author Contributions: Conceptualization, J.M., S.H., S.-J.K. and M.H.; methodology, J.M., S.H. and M.H.; software, S.H., J.M.; validation, J.M. and S.H.; formal analysis, J.M., S.-J.K. and M.H.; investigation, J.M. and S.H.; resources, J.M. and S.H.; data curation, J.M. and S.H.; writing-original draft preparation, J.M.; writing-review and editing, J.M., S.-J.K. and M.H.; visualization, M.H.; supervision, S.-J.K. and M.H.; project administration, M.H.; funding acquisition, M.H.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. CompTIA. Sizing up the Internet of Things. Available online: https://www.comptia.org/resources/sizing-up-the-internet-of-things (28 August 2015).
2. Kotsiou, V.; Papadopoulos, G.Z.; Chatzimisios, P.; Theoleyre, F. Adaptive multi-channel offset assignment for reliable IEEE 802.15.4 TSCH networks. In Proceedings of the 2018 Global Information Infrastructure and Networking Symposium (GIIS), Thessaloniki, Greece, 23–25 October 2018.
3. Chincoli, M.; den Boef, P.; Liotta, A. Cognitive channel selection for wireless sensor communications. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017.
4. IEEE 802. *802.15.4e-2012: IEEE Standard for Local and Metropolitan Area Networks-Part15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1:MAC Sub-Layer*; IEEE Std: New York, NY, USA, 2016.
5. Nguyen, D.D.; Nguyen, H.X.; White, L.B. Reinforcement Learning With Network-Assisted Feedback for Heterogeneous RAT Selection. *IEEE Trans. Wireless Commun.* **2019**, *19*, 6062–6076.
6. Wu, C.-M.; Wu, M.-S.; Yang, Y.-J.; Sie, C.-Y. Cluster-based distributed MAC protocol for multi-channel cognitive radio ad hoc networks. *IEEE Access* **2019**, *7*, 65781–65796.
7. Jang, S.-J.; Han, C.-H.; Lee, K.-E.; Yoo, S.-J. Reinforcement learning-based dynamic band and channel selection in cognitive radio ad-hoc networks. *Eurasip J. Wirel. Commun. Netw.* **2019**, *2019*, 131.
8. Robbins, H. Some aspects of the sequential design of experiments. *Bull. Am. Math. Soc.* **1952**, *58*, 527–535.
9. Kim, S.-J.; Aono, M.; Nameda, E. Efficient decision-making by volume-conserving physical object. *New J. Phys.* **2015**, *17*, 083023.
10. Kim, S.-J.; Aono, M.; Hara, M. Tug-of-war model for the two-bandit problem: Nonlocally-correlated parallel exploration via resource conservation. *BioSystems* **2010**, *101*, 29–36.
11. Kim, S.-J.; Aono, M.; Nameda, E.; Hara, M. *Amoeba-Inspired Tug-of-War Model: Toward a Physical Implementation of an Accurate and Speedy Parallel Search Algorithm*; Complex Communication Sciences (CCS), IEICE: Okinawa, Japan, 2011; pp. 36–41.
12. Kim, S.-J.; Aono, M. Amoeba-inspired algorithm for cognitive medium access. *NOLTA* **2014**, *5*, 198–209.
13. Kuroda, K.; Kato, H.; Kim, S.-J.; Naruse, M.; Hasegawa, M. Improving throughput using multi-armed bandit algorithm for wireless LANs. *NOLTA* **2018**, *9*, 74–81.
14. IEEE 802. *802.15.4g-2012: IEEE Standard for Local and Metropolitan Area Networks-Part15.4:Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 3: Physical Layer (PHY) Specifications for Low-Data-Rate, Wireless, Smart Metering Utility Networks*; IEEE Std: New York, NY, USA, 2016.
15. Sutton R.; Barto, A. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
16. Vermorel, J.; Mohri, M. Multi-armed bandit algorithms and empirical evaluation. In Proceedings of the 16th European Conference on Machine Learning, Porto, Portugal, 3–7 October 2005; pp. 437–448.
17. Lai, T.L.; Robbins, H. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* **1985**, *6*, 4–22.
18. Auer, P.; Cesa-Bianchi, N.; Fischer, P. Finite-time analysis of the multiarmed bandit prblem. *Mach. Learn.* **2002**, *47*, 235–256.
19. LAPIS Semiconductor. *Lazurite920mhz(Q504H) Datasheet*; LAPIS: Shinyokohama, Japan, 2010.
20. IEEE. Available online: https://iot.ieee.org/newsletter/january-2017/internet-of-things-for-buildings-that-make-life-safe-and-secure.html (10 January 2017).