2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-7177-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICDCS54860.2022.00073

Defending against Cross-Technology Jamming in Heterogeneous IoT Systems

Sihan Yu*, ChunChih Lin*, Xiaonan Zhang†, Linke Guo*
*Clemson University, Clemson, SC, USA
†Florida State University, Tallahassee, FL, USA
{sihany, chunchi, linkeg}@clemson.edu, xzhang@cs.fsu.edu

***Abstract*—The wide deployment of IoT devices has resulted in a critical shortage of spectrum resources. Many IoT devices coexist on the same spectrum band, where the network performance is always degraded. As a promising solution, the Cross-Technology Communication (CTC) enables the direct communication among heterogeneous IoT devices. Unfortunately, the emerging cross-technology attacks have demonstrated their high success rates in terms of spoofing the end IoT devices or jamming the communication channels. In this paper, we investigate a novel cross-technology jamming issue for a distributed heterogeneous IoT system. Compared with traditional jamming methods, the cross-technology jammer has a much higher jamming power, wider jamming bandwidth, and stronger stealthiness, all of which deserve a complete re-thinking of defensive mechanisms. Therefore, we propose a hybrid anti-jamming scheme that jointly considers frequency hopping and power control techniques. Specifically, we model the anti-jamming process as a Markov Decision Process (MDP) and adopt Deep Q-Network (DQN) to find the optimal strategy. Extensive real-world experiments show that the goodput (payload data) of our anti-jamming scheme can achieve up to 2X and 1.39X than the passive and random anti-jamming approaches, respectively. In particular, our anti-jamming scheme provides 78% of goodput with the presence of a cross-technology jammer, outperforming existing passive and random anti-jamming scheme designs at 37.6% and 54.1%.**

***Index Terms*—Cross-Technology Jamming, Anti-jamming, Deep Q-Network, Markov Decision Process, Internet of Things**

I. Introduction

Recent years have witnessed the increasing deployment of Internet-of-Thing (IoT) in our daily life, including smart home, healthcare, smart city, and smart manufacturing. According to Statista, the number of IoT devices worldwide will be 38.6 billion by 2025 [1] for supporting a broader range of future applications. Current mainstream IoT protocols, such as Wi-Fi, ZigBee, and Bluetooth/Bluetooth Low Energy (BLE), heavily overlap on the 2.4GHz Industrial, Scientific, and Medical (ISM) bands. Apparently, the coexistence of a large number of heterogeneous IoT devices brings significant challenges for coordinating each transmission link given limited spectrum resources. Although many link layer protocols have been widely adopted, such as CSMA/CA, those schemes become less effective in handling heterogeneous IoT devices with a dense deployment, e.g., future warehouses for smart manufacturing. The coexistence of heterogeneous IoT devices may not only increase the chance of experiencing cross-technology interference (CTI) [2], [3], but also incur serious security concerns (e.g., jamming attacks or spoofing attacks). Due to the protocol incompatibility, existing defensive methods will fail to detect and mitigate them.

As one of the most promising communication paradigms, the Cross-Technology Communication (CTC) [4], [5] has achieved the direct communication among IoT devices with different protocols. Although it potentially relieves the long-term coexistence problem, it brings significant threats to the current IoT systems, especially for the crowded 2.4GHz ISM band. For example, attackers can go far beyond just simply sending interference signals [6]–[9], instead, sending a CTC spoofing signal can bypass the victim's security protocol and finally compromise the entire system. Recent works [10]–[12] elaborate that using CTC signals can spoof end IoT devices from a further distance, and severely degrade the packet delivery ratio (PDR). Instead of just spoofing an individual IoT device, in this paper, we investigate a crucial yet less-investigated problem, i.e., *how to defend against cross-technology jamming attacks in IoT networks?* Although there have been a few works on mitigating cross-technology jamming [11], [13], [14], the proposed schemes either partially recover the collided/interfered packets or only have a marginal gain on increasing PDR ($< 2\%$), falling far short of fundamentally defending against the jamming attack.

In this paper, we consider a Wi-Fi-enabled jammer that intends to attack a ZigBee network. Compared with a conventional ZigBee jammer, this cross-technology jammer has stronger stealthiness, wider jamming bandwidth, and higher attacking power, all of which bring significant challenges for anti-jamming scheme design. Specifically, we focus on a smart jammer that can sweep all available channels and choose the suitable power to attack. Meanwhile, the jammer has a faster sweep cycle and can jam up to 4 consecutive ZigBee channels, leading to a lower success rate of transmission if simply adopting existing anti-jamming schemes.

To defend against the cross-technology jamming (CTJ) attack, we employ both the frequency hopping and power control to develop the anti-jamming strategy. The anti-jamming process is modeled as a Markov Decision Process (MDP), in which new states other than conventional anti-jamming schemes [15] are introduced. Then, we adopt Deep Q-Network

The work of L. Guo was supported by National Science Foundation under grant IIS-1949640 and CNS-2008049.

2575-8411/22/$31.00 ©2022 IEEE
DOI 10.1109/ICDCS54860.2022.00073

(DQN) to find the optimal defense strategy about how to select the channel and signal strength properly. The main contributions of this paper are as follows:

- We identify a new but real CTJ attack in the heterogeneous IoT environment, where a Wi-Fi-emulated ZigBee signal is able to achieve up to 4 times higher jamming performance compared with a conventional Zigbee jammer.
- We model the anti-jamming process as an MDP process and prove the existence of the optimal strategy. To better understand the competition process, we also study the variation trend of the optimal strategy with various parameters.
- We further adopt DQN to find the optimal defense strategy. Based on extensive experiments, we demonstrate that our proposed scheme can successfully defend against CTJ and maintain a high success rate of transmission.

The rest of the paper is organized as follows. In Section II, we introduce the motivation and feasibility of CTJ. Then, we model the competition process into an MDP and adopt DQN to find the optimal defense strategy in Section III. Section IV shows the performance of our defense strategy, followed by the state-of-the-art discussion in Section V. Finally, Section VI concludes the paper.

II. Preliminaries and Motivation

In this section, we first briefly introduce the malicious CTJ signal generation process, and further elaborate on how the signal can compromise the ZigBee network performance.

A. Generating the Jamming Signal

1) Cross-technology Signal Emulation: Existing approaches [4], [5], [16] have been proposed to emulate ZigBee signals using the Wi-Fi device. As shown in Fig. 1, the emulation process can be regarded as the inverse process of the WiFi physical layer, because the jammer needs to know the bit stream that can generate the desired waveforms. However, existing designs fail to provide precise emulation because the 64-QAM constellation diagram is usually not fully utilized.

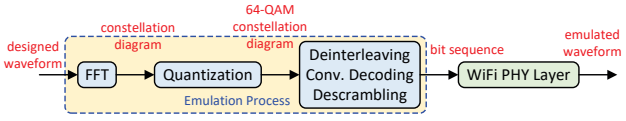


Fig. 1. Emulation Process

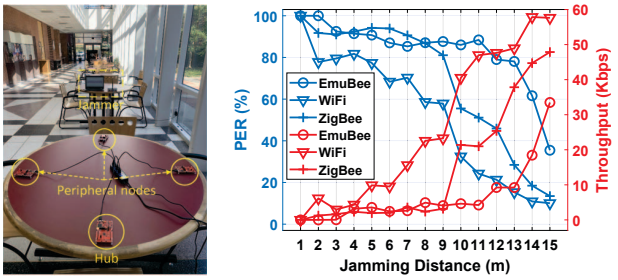
For this work, we will modify the quantization process to make better use of the 64-QAM constellation diagram. With this design, the emulated waveforms will be more similar to the designed waveforms. In what follows, we formulate the above quantization process as an optimization problem in order to minimize the distance between the 64-QAM points and the corresponding constellation points of the ZigBee signals, as shown in (1) and (2),

$$E(\alpha) = \sum_{j=1}^{M} \min\{(\alpha P_i - P_j)^2 | i \in [1, 64]\} \quad (1)$$

$$\alpha = \underset{\alpha}{\operatorname{argmin}}\ E(\alpha) \quad (2)$$

in which $E(\alpha)$ depicts the overall quantization error, and α is a scalar to scale the size of 64-QAM constellation diagram. P_i is a complex number that denotes the i-th constellation point among 64 predefined constellation points, whereas P_j is the j-th constellation point that will be used to emulate signals. M is the total number of constellation points. Eq. (2) depicts the objective of our optimization, which is to find a proper α that can minimize (1). Since $E(\alpha)'' = \sum_{j=1}^{M} 2P_i^2 > 0$, $E(\alpha)$ in (1) is a convex function and has the global minimum, which can be acquired in $\mathcal{O}(M \log M)$ for M FFT points if using the binary search algorithm.

2) Experimental Validation: To verify our idea, we conduct an experiment to test the jamming effect of different signals. We use four TI CC26X2R1 LaunchPads (ZigBee mode) to build a star-like ZigBee IoT network (Fig. 2(a)), in which one of them acts as the hub, the rest act as peripheral nodes. The Listen-Before-Talk (LBT) mechanism is adopted to avoid collisions. Then, we use a USRP N210 as the jammer. For each time, the jammer sends different jamming signals (i.e., WiFi, ZigBee, and Emulated ZigBee) at different distances.



(a) Experiment Scenario (b) Jamming Effect of Different Signals

Fig. 2. Effect Verification

We evaluate the throughput and packet error rate (PER) of this network under different types of jamming attacks, as shown in Fig. 2(b). It can be seen that with the increase of jamming distance, the PER continuously decreases while the throughput increases. More importantly, in most cases, the rank in terms of the jamming effect is as follows: Emulated ZigBee (EmuBee)>ZigBee>WiFi. The main reason is that EmuBee is generated by a WiFi device, which has a higher signal strength than that of ZigBee devices. This superiority is more significant when the jamming distance is long (i.e., $\geq 10m$). On the other hand, the WiFi jammer is the weakest one because the IoT network regards it as noises and the direct-sequence spread spectrum (DSSS) mechanism of ZigBee devices has a very good effect on dealing with noises. Another advantage of using EmuBee as the jamming signals is that the EmuBee has a better stealthiness, because it is not necessary to obey the format of ZigBee packets, which will be more difficult to be detected by ZigBee devices. Specifically, a complete ZigBee packet consists of preamble (`0x00000000`), start-of-packet delimiter (`0x7A`), PHY header (1 octets) and PHY payload, as shown in Fig. 3. If a ZigBee packet does not obey the above format, the ZigBee receiver will not able to get

useful information from it, although it still tries to decode it. For example, if a ZigBee packet only has the preamble (i.e. the delimiter and rest part are missing), the ZigBee receiver will process it into the decoding state. However, over a period of time, nothing can be decoded. Meanwhile, the hardware resource is being occupied and cannot be used to process other packets. Therefore, using EmuBee to launch a jamming attack is very hard to be detected by ZigBee receivers.

Preamble	Start-of-packet Delimiter	PHY Header	PHY Srvice Data Unit (PSDU)
0x00000000	0x7A	1B	<= 127B

Fig. 3. ZigBee Packet Format

B. Motivation of Cross-Technology Jamming

According to the above experimental result, CTJ outperforms the conventional jamming attacks using the same-protocol jamming signals. Specifically, CTJ has the following advantages that make it as an ideal attacker in the heterogeneous IoT system.

- **Higher jamming power**. EmuBee adopts the Wi-Fi protocol to launch the attack, whose RF power can be up to 100mW. However, ZigBee concerns more about energy efficiency, whose RF power can be as low as 1mW. For the same reason, the attack range of WiFi is also wider.
- **Wider jamming bandwidth**. The bandwidth of a WiFi channel (i.e. 20MHz) is 10 times more than that of a ZigBee channel (i.e. 2MHz). According to the spectral overlap of WiFi and ZigBee, a WiFi jammer can scan and jam up to 4 ZigBee channels at a time. Thus, a wider bandwidth indicates that the jammer can jam more ZigBee devices as well as find them faster.
- **Stronger stealthiness**. Based on the analysis of the stealthiness, EmuBee uses the ZigBee waveform but does not need to follow the ZigBee packet format, resulting in the "meaningless" decoding at the victim IoT device. Compared with traditional jammers where the jamming signal is either standard ZigBee signal or noise, EmuBee can fool the victim IoT device for decoding, and thus is hidden from being detected as a jammer.

C. Adversarial Model

1) Cross-Technology Jammer's Capability: The jammer (Jx) adopts the time-slotted frequency-sweeping to find the victim. It can sweep and jam m consecutive channels at a time ($m \leq K$, K is the number of channels that can be used). Jx will send EmuBee signals only when the victim is using the channel, thus, the victim is not able to detect Jx by measuring the received signal strength. The strength of the jamming signal is adjustable with different power levels $P_J = \{p_1^J, p_2^J, ..., p_L^J\}$. For each time, Jx chooses a power level to send jamming signals. We define two modes for the Jx: (1) high-performance mode: in this mode, the major objective of the Jx is to jam transmission as much as possible. Therefore, Jx always picks the largest power level to send jamming signals; (2) hidden mode: in this mode, the major objective of the Jx is to avoid being perceived by the victim. It should not jam too much transmission, which may lead to the victim no longer using this channel. Therefore, Jx randomly picks a power level to send jamming signals. The Jx is able to know whether the jamming is successful by monitoring whether the victim is still using the channel at the beginning of each time slot. Alternatively, the Jx can passively listen to the feedback information, such as ACK/NACK message.

For the victim, they also adopt the time-slotted working mode. At the beginning of each time slot, the hub decides which channel and power level will be used, and then sends this information to peripheral nodes. During the process of running, peripheral nodes will send data to the hub. If the power level of Jx is higher than that of peripheral nodes, the peripheral nodes will be not able to transmit data correctly.

2) Attacking Process: Once the Jx initiates, it will begin to sweep the channels with the speed of m channels/time-slot, i.e., 4 ZigBee channels for Wi-Fi attacker using EmuBee. On the 2.4GHz frequency band, there are 16 available ZigBee channels in total, so the Jx only needs to spend $\lceil 16/4 \rceil = 4$ time slots (a.k.a. *sweep cycle*) on sweeping the whole channels to find the victim.

If the victim IoT device is found, Jx no longer sweeps other channels, instead, it begins to jam the victim's channel. The victim will then notice the throughput drop. Once the error rate exceeds a certain threshold, the victim will decide to hop to another channel or select a higher power to transmit data. In the case when the hub cannot contact peripheral nodes using the current channel, we assume the existence of a control channel for negotiating the communication channel. If the Jx finds that the victim no longer uses the current channel, the Jx will continue to sweep other channels for seeking the next opportunity of launching an attack.

III. Deep Learning-based Anti-Jamming Scheme

To defend against the proposed jamming attack, we propose a hybrid anti-jamming approach that jointly considers frequency hopping and power control mechanisms.

A. Problem Formulation

First of all, we formulate the interaction between the Jx and the Zigbee IoT devices as an MDP, in which the future state only depends on the current state and action.

1) State Space: We define the state space as

$$X = \{1, 2, ..., \lceil K/m \rceil - 1, T_J, J\}, \tag{3}$$

where $n \in \{1, 2, ..., \lceil K/m \rceil - 1\}$ denotes that the legitimate ZigBee transmitter (Tx) has been continuously successful in transmitting data on the current channel for n time slots. The Jx sweeps through the K channels and can jam up to m ($m \leq K$) consecutive channels at once. T_J denotes that the transmission is jammed but not successfully (i.e., the transmission is still successful, because the jamming signal power is not enough to completely compromise the communication). We use J to represent that the current transmission is completely jammed.

2) Action Space: The action space for any state in X is as follows:

$$A = \{(s, p_1^T), ..., (s, p_M^T), (h, p_1^T), ..., (h, p_M^T)\} \quad (4)$$

where $(s, p_i^T),\ i \in \mathcal{M}$ denotes the action that Tx stays on the current channel with the transmission power p_i. Similarly, $(h, p_i^T),\ i \in \mathcal{M}$ is the action that Tx hops to a new channel and uses p_i for transmission.

3) Immediate Reward: The immediate reward of moving from the state x to x' with an action a is defined as below,

$$U(x,a,x')=\begin{cases} -L_{p_i^T} - L_J & a=(s,p_i^T), x'=J \\ -L_{p_i^T} & a=(s,p_i^T), x' \in X \backslash J \\ -L_{p_i^T} - L_J - L_H & a=(h,p_i^T), x'=J \\ -L_{p_i^T} - L_H & a=(h,p_i^T), x' \in X \backslash J \end{cases} \quad (5)$$

where $L_{p_i^T}$ denotes the loss of using power p_i^T to send packets; L_J is the loss due to a successful jamming; and L_H is used to quantify the loss due to the frequency hopping (since frequency hopping will lead to packet loss or increased latency).

4) Transition Probabilities: Let $P(x'|x,a)$ denote the transition probability to x' given that the current state is x, the Tx chooses action $a \in A$. Then, the state transition mainly involves the following cases:

Case 1: The Tx is not jammed in the current time slot, after choosing (s, p_i^T) as the action, the Tx is still not jammed in the next time slot. In this case, we have

$$P(n+1|n,s,p_i^T) = 1 - \frac{1}{\lceil K/m \rceil - n} \quad (6)$$

where $n \in \{1, 2, ..., \lceil K/m \rceil - 2\}$.

Case 2: Similar to **Case 1**, the Tx is jammed unsuccessfully/successfully in the next time slot can be expressed as

$$P(T_J|n,s,p_i^T) = \frac{1}{\lceil K/m \rceil - n} \times P(p_i^T > \tau) \quad (7)$$

$$P(J|n,s,p_i^T) = \frac{1}{\lceil K/m \rceil - n} \times P(p_i^T < \tau) \quad (8)$$

where $n \in \{1, 2, ..., \lceil K/m \rceil - 1\}$.

Case 3: Given the Tx is not jammed in the current time slot, it is also not jammed in the next time slot due to taking (h, p_i^T) as the action can be expressed as

$$P(1|n,h,p_i^T) = 1 - \frac{\lceil K/m \rceil - n - 1}{(\lceil K/m \rceil - 1)(\lceil K/m \rceil - n)} \quad (9)$$

where $n \in \{1, 2, ..., \lceil K/m \rceil - 1\}$.

Case 4: Similar to **Case 3**, the Tx is jammed unsuccessfully/successfully in the next time slot is represented by

$$P(T_J|n,h,p_i^T) = \frac{\lceil K/m \rceil - n - 1}{(\lceil K/m \rceil - 1)(\lceil K/m \rceil - n)} \times P(p_i^T > \tau) \quad (10)$$

$$P(J|n,h,p_i^T) = \frac{\lceil K/m \rceil - n - 1}{(\lceil K/m \rceil - 1)(\lceil K/m \rceil - n)} \times P(p_i^T < \tau) \quad (11)$$

where $n \in \{1, 2, ..., \lceil K/m \rceil - 1\}$.

Case 5: Different to the previous four cases, the Tx is jammed unsuccessfully/successfully in the current time slot. After choosing (s, p_i^T) as the action, the Tx is jammed unsuccessfully/successfully in the next time slot. We have

$$P(T_J|x,s,p_i^T) = P(p_i^T > \tau) \quad (12)$$

$$P(J|x,s,p_i^T) = P(p_i^T < \tau) \quad (13)$$

where $x \in \{T_J, J\}$.

Case 6: Similar to **Case 5**, the Tx is not jammed in the next time slot after taking (h, p_i^T) as the action is expressed as

$$P(1|x,h,p_i^T) = 1 \quad (14)$$

where $x \in \{T_J, J\}$.

B. Analysis of MDP

1) Existence of Optimal Policies: Before trying to solve the MDP, it is of great importance to prove the existence of the optimal policy.

Theorem III.1. *For any finite MDP, there exists an optimal policy π^* such that it is better than or equal to every other possible policy π.*

Proof. See the proof in the Appendix. □

2) Calculation Method of Optimal Policies: For an MDP, a policy $\pi(a|x)$ is the probability of taking action a at state x. The state-value function of a state x under a policy π is defined as

$$V_\pi(x) = \sum_a \pi(a|x) Q_\pi(x,a) \quad (15)$$

Similarly, the action-value function of taking action a in state x is defined as

$$Q_\pi(x,a) = U(x,a) + \gamma \sum_{x'} P(x'|x,a) V_\pi(x') \quad (16)$$

where $\gamma \in [0,1]$ is the discount factor that quantifies how much importance we give for future rewards. When solving an MDP, we need to find the optimal value functions, which are defined as follows:

$$V_*(x) = \max_\pi V_\pi(x) \quad (17)$$

$$Q_*(x,a) = \max_\pi Q_\pi(x,a) \quad (18)$$

The optimal policy is referred to as the optimal value functions. To find the optimal policy, we pick the action that gives us the maximum $Q_*(x,a)$,

$$\pi_*(a|x) = \begin{cases} 1 & \text{if } a = \arg\max_a Q_*(x,a) \\ 0 & otherwise \end{cases} \quad (19)$$

Thus, the following results can be derived.

$$\begin{aligned} V_*(x) &= \max_a Q_*(x,a) \\ &= \max_a U(x,a) + \gamma \sum_{x'} P(x'|x,a) V_*(x') \end{aligned} \quad (20)$$

$$Q_*(x,a) = U(x,a) + \gamma \sum_{x'} P(x'|x,a) \max_{a'} Q_*(x',a') \quad (21)$$

The above equation is called Bellman optimality equation [17], which is a recursive equation. We can get the optimal policy as well as the optimal value functions by solving this equation.

3) Features of MDP: The aforementioned process is a general method to solve an MDP. To better understand our MDP model, we also study our own features, i.e., how the action-value function as well as the optimal policy vary with various influence factors. Our constructed MDP has the following features:

Lemma III.2. *$Q\left(n,(s,p_i^T)\right)$ is a decreasing function on $n=1,2,...,\lceil K/m\rceil-1$.*

Proof. We can prove this by showing $Q_*(n,(s,p_i^T)) < Q_*(n-1,(s,p_i^T))$.

$$\begin{aligned}&Q_*(n,(s,p_i^T))-Q_*(n-1,(s,p_i^T))\\&=\left[U(n,a)+\gamma\sum_{n'}P(n'|n,a)\max_{a'}Q_*(n',a')\right]\\&-\left[U(n-1,a)+\gamma\sum_{(n-1)'}P((n-1)'|n-1,a)\max_{a'}Q_*((n-1)',a')\right]\end{aligned}\tag{22}$$

From (5), (6), (7), (8), it is easy to show

$$\mathbb{E}[U(n,(s,p_i^T))]=-L_p-L_J\tfrac{P(p_i^T<\tau)}{\lceil K/m\rceil-n}\tag{23}$$

i.e., $U(n,(s,p_i^T))$ is decreasing in n. Thus, $U(n,a) < U(n-1,a)$.

When $n=\lceil K/m\rceil-1$,

$$\begin{aligned}&\max_{a'}Q_*(n',a')=\max_{a'}\{Q_*(J,a'),Q_*(T_J,a')\}\\&\le\max_{a'}\{Q_*(J,a'),Q_*(T_J,a'),Q_*(n,a')\}=\max_{a'}Q_*((n-1)',a')\end{aligned}$$

Thus, $Q_*(n,(s,p_i^T)) < Q_*(n-1,(s,p_i^T))$.

Similarly, we can also show $Q_*(n-1,(s,p_i^T)) < Q_*(n-2,(s,p_i^T))$. This process can go all the way up to $Q_*(2,(s,p_i^T)) < Q_*(1,(s,p_i^T))$, leading to a conclusion that $Q_*(n,(s,p_i^T))$ is a strictly decreasing function with the increase of n. □

Lemma III.3. *$Q\left(n,(h,p_i^T)\right)$ is an increasing function on $n=1,2,...,\lceil K/m\rceil-1$.*

Proof. We can prove lemma 2 by showing $Q_*(n,(h,p_i^T)) > Q_*(n-1,(h,p_i^T))$.

From (5), (9), (10), (11), it is easy to show

$$\mathbb{E}[U(n,(h,p_i^T))]=-L_p-L_H-L_J\tfrac{P(p_i^T<\tau)(\lceil K/m\rceil-n-1)}{(\lceil K/m\rceil-1)(\lceil K/m\rceil-n)}\tag{24}$$

i.e., $U(n,(h,p_i^T))$ is increasing in n. Thus, $U(n,a) > U(n-1,a)$.

In addition, we have

$$\begin{aligned}&\max_{a'}Q_*(n',a')=\max_{a'}\{Q_*(J,a'),Q_*(T_J,a'),Q_*(1,a')\}\\&=\max_{a'}Q_*((n-1)',a')\end{aligned}$$

Thus, $Q_*(n,(h,p_i^T)) > Q_*(n-1,(h,p_i^T))$, i.e., $Q_*(n,(h,p_i^T))$ is a strictly increasing function with the increase of n. □

Based on lemma III.2 and lemma III.3, the optimal policy has the following structure stated in Theorem III.4.

Theorem III.4. *The optimal policy can be characterized by a threshold $n^*\in\{1,2,...,\lceil K/m\rceil\}$, i.e.,*

$$a^*=\pi^*(n)=\begin{cases}(s,p_i^T) & if\ n<n^*\\(h,p_i^T) & otherwise\end{cases}\tag{25}$$

Proof. Since $Q\left(n,(s,p_i^T)\right)$ is decreasing and $Q\left(n,(h,p_i^T)\right)$ is increasing, there must exist an intersection point between them except two extreme cases. One is $Q(\lceil K/m\rceil-1,(s,p_i^T)) \ge Q(\lceil K/m\rceil-1,(h,p_i^T))$, where we can set n^* as $\lceil K/m\rceil$. The other is $Q(1,(s,p_i^T)) \le Q(1,(h,p_i^T))$, where we can set n^* as 1. □

Theorem III.5. *The threshold n^* decreases with the increase of L_J, and increases with the increase of L_H or $\lceil K/m\rceil$.*

Proof. From (23) and (24), it is easy to show

$$U(n,(s,p_i^T))-U(n-1,(s,p_i^T))=-\tfrac{L_JP(p_i^T<\tau)}{(\lceil K/m\rceil-n)(\lceil K/m\rceil-n+1)}\tag{26}$$

$$U(n,(h,p_i^T))-U(n-1,(h,p_i^T))=\tfrac{L_JP(p_i^T<\tau)}{(\lceil K/m\rceil-1)(\lceil K/m\rceil-n)(\lceil K/m\rceil-n+1)}\tag{27}$$

Substitute (26) in to (22), we can get that $Q_*(n,(s,p_i^T)) - Q_*(n-1,(s,p_i^T))$ decreases with the increase of L_J. In other words, the function graph of $Q_*(n,(s,p_i^T))$ will move down with the increase of L_J. From (27), we can infer that the function graph of $Q_*(n,(h,p_i^T))$ will move up with the increase of L_J. Therefore, their interaction point will move left, i.e., the threshold n_* decreases with the increase of L_J.

Similarly, we can also find that the function graph of $Q_*(n,(s,p_i^T))$ ($Q_*(n,(h,p_i^T))$) will move up (down) with the increase of $\lceil K/m\rceil$. Thus, the threshold n_* increases with the increase of $\lceil K/m\rceil$.

From (23) and (24), we know that $U(n,(s,p_i^T))$ is not impacted by L_H, whereas $U(n,(h,p_i^T))$ decreases with the increase of L_H. Thus, the interaction point of $Q_*(n,(s,p_i^T))$ and $Q_*(n,(h,p_i^T))$ will move right, i.e., the threshold n_* increases with the increase of L_H. □

From the above analysis, we can conclude that there exists a turning point, before which (s,p_i^T) should be chosen as the action, and after that (h,p_i^T) will be the optimal policy. Meanwhile, the turning point can be adjusted by setting L_J, L_H and $\lceil K/m\rceil$ properly.

C. DQN based Anti-jamming Scheme

Although the optimal policy can be acquired by solving the MDP, the derived result cannot be directly used in an IoT network, because the Tx does not exactly know its current state x. The state x not only depends on the Tx, but also depends on the Jx. However, in the practical setting, the Jx is hardly able to be synchronized with the Tx. Therefore, the derived result is idealized. We still need to get an adaptive frequency hopping as well as a power control scheme, which has the function of self-correction in a real-time way according to the external environment. Reinforcement learning (RL) is a suitable technique that can be used to acquire an optimal communication policy via trial-and-error. The advantage of RL is that it does not depend on any model. Even if we do

not know the model of the competition process, we can still use RL to get an optimal policy.

For this work, we use a deep Q-network (DQN) to acquire an optimal communication policy. Compared with other RL techniques (such as Q-learning), the learning speed of DQN will not suffer from the curse of high-dimensionality, i.e., the convergence time will not significantly increase with the dimension of state space and action space. As shown in Fig. 4, we adopt a 4-layer deep neural network (DNN) to train the anti-jamming scheme. Note that other neural networks architectures (e.g. RNN) can also be adopted, since our scenario is not a large-scale problem, adopting the simple and fully-connected DNN is sufficient to solve it. The input layer has $3 \times I$ neurons, which correspond to the state (i.e., success or failure) and action (i.e., channel and power level) of the Tx in previous I time slots because these three indexes are observable to the victim. The output layer has $C \times P_L$ neurons, where C, P_L are the number of available channels and power levels of the Tx, respectively. We also adopt 2 fully connected layers and use ReLU as the activation function, since 2 fully connected layers are sufficient to solve nonlinear problems. To avoid that the DQN stays at a local maximum, we choose the communication policy based on the ε-greedy algorithm. Specifically, the optimal communication policy with the highest Q-value is chosen with a high probability $1-\varepsilon$, and other feasible strategies are chosen with a small probability $\varepsilon/(C \times P_L - 1)$.

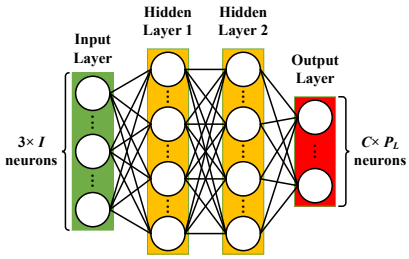


Fig. 4. Neural Network Architecture of DQN

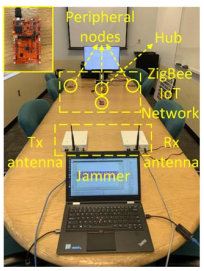


Fig. 5. Experiment Setting

IV. Performance Evaluation

To evaluate the anti-jamming scheme design, we will conduct both simulations and real-world field experiments.

A. Experiment Settings

1) Simulation Settings: We use Matlab as the tool to train and evaluate the DQN. To best describe the WiFi and ZigBee coexistence scenario, (1) we set the sweep cycle to 4 (i.e., $\lceil K/m \rceil = 4$), because a WiFi device is able to scan all available ZigBee channels in 4 time slots; (2) we provide 10 different power levels for the victim and the attacker, respectively, i.e., $L^T_{p_i} \in [6, 15]$, $L^J_{p_i} \in [11, 20]$. It can be seen that the max power of Jx is higher than that of Tx, which ensures that Jx is able to jam Tx. For the sake of simplicity, we believe that the transmission will be successful if $L^T_{p_i} \geq L^J_{p_i}$; (3) $L_H = 50$, i.e., the loss of frequency hopping (FH) is higher than that of adopting power control (PC), because conducting FH will take some time for negotiation and affect the throughput; (4) $L_J = 100$, because the loss of being jammed should be significantly higher than that of FH and PC, so that the system will be encouraged to take measures to avoid being jammed.

TABLE I
Evaluation Metrics

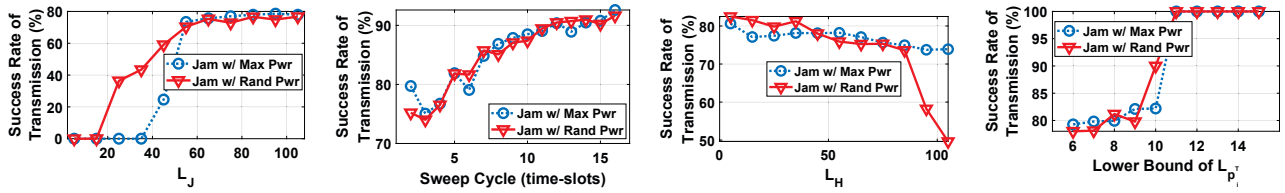
Evaluation Metrics	Description
Success rate of transmission (S_T)	The proportion of time slots that transmit data successfully
Adoption rate of FH (A_H)	The ratio of time slots adopting FH to the total number of time slots
Success rate of FH (S_H)	The ratio of time slots that transmit data successfully due to adopting FH to the number of time slots that adopting FH
Adoption rate of PC (A_P)	The ratio of time slots adopting PC to the total number of time slots
Success rate of PC (S_P)	The ratio of time slots that transmit data successfully due to adopting PC to the number of time slots that adopting PC

The detailed evaluation metrics are shown in Table I. The S_T denotes the proportion of time slots that can be used to transmit data successfully. The A_H evaluates the proportion of time slots that adopt FH. A good anti-jamming strategy should adopt FH as few as possible because FH is time-consuming. Similarly, the S_H evaluates the proportion of useful FH, because some of the FHs are adopted for preventative purposes, which are actually unnecessary. Meanwhile, A_P, S_P are similar to A_H, S_H, respectively. We will measure the above-mentioned metrics in two scenarios, where the Jx adopts either (1) the max power mode or (2) the random power mode. For each time, the experiment lasts for 20000 time slots to get the average value.

2) Field Experiments Settings: For the field experiment, two USRP N210 and multiple TI CC26X2R1 LaunchPads (ZigBee mode) are used to verify the effectiveness and efficiency of our scheme, as shown in Fig. 5. The targeted ZigBee IoT network is composed of 4 nodes, in which one node acts as the hub and the others act as peripheral nodes. The trained DQN is loaded into the hub to decide the channel and power level that should be used in the next time slot. Then, the hub will notify peripheral nodes of the FH and PC information in advance. The transmitted information can be encrypted to prevent eavesdropping. If they cannot contact each other (e.g., the current channel is jammed unexpectedly), we also assign a control channel to exchange information. Two USRPs are used as the jammer to send jamming signals and eavesdrop on the current channel, respectively. Note that the Tx and Rx function can also be integrated to one USRP, but the jamming effect will degrade because it has to switch between two modes frequently.

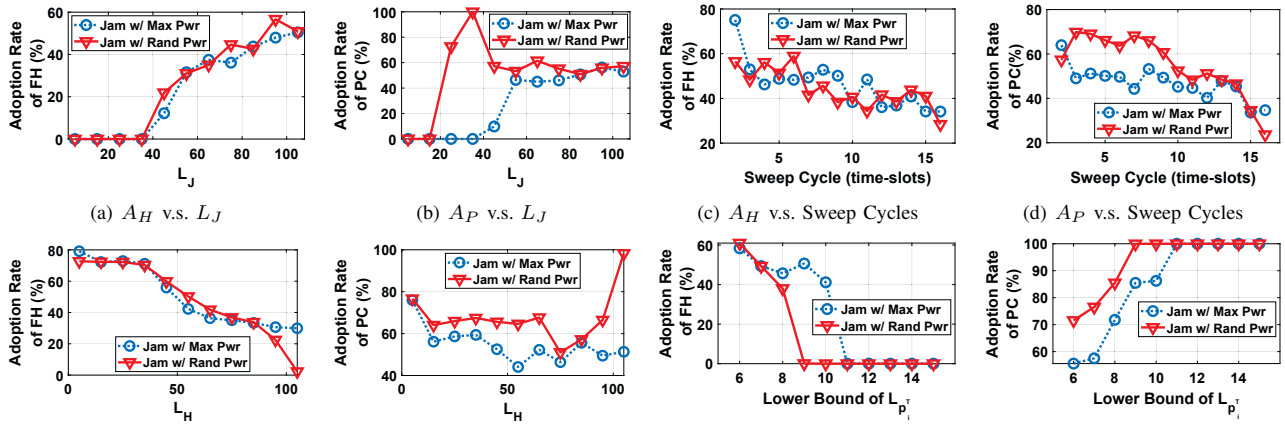
B. Training of DQN

We use more than 120000 data blocks from historical information to train the DQN, in which each data block contains the channel, power level and state (success or failure) information of the Tx. The training process lasts on average of 20min unless the training goal has been achieved in advance (i.e., the average reward reaches a certain threshold). The



(a) S_T v.s. L_J (b) S_T v.s. Sweep Cycles (c) S_T v.s. L_H (d) S_T v.s. $L_{p_i^T}$

Fig. 6. Success Rate of Transmission



(a) A_H v.s. L_J (b) A_P v.s. L_J (c) A_H v.s. Sweep Cycles (d) A_P v.s. Sweep Cycles

(e) A_H v.s. L_H (f) A_P v.s. L_H (g) A_H v.s. $L_{p_i^T}$ (h) A_P v.s. $L_{p_i^T}$

Fig. 7. Adoption Rate of FH and PC

training results are a series of matrices, which contain 10664 float numbers with 42.7KB memory. Those parameters will be loaded to the IoT devices before the experiment starts.

C. Simulation-based Experimental Results

1) Success Rate of Transmission: The success rate of transmission (S_T) is the most important metric to evaluate the effectiveness of an anti-jamming approach. In a time-slotted system, S_T can be defined as the proportion of time slot numbers that can be used to transmit data successfully. Various parameters can influence the S_T during the training process. In what follows, we mainly study the impact of L_J, sweep cycle, L_H and $L_{p_i^T}$ on S_T.

Impact of L_J. A larger L_J will let the agent be more active in mitigating jamming attacks. Fig. 6(a) shows that the S_T increases with the increase of L_J. When the $L_J \leq 15$ (note that 15 is the upper limit of $L_{p_i}^T$), the success rate is maintained at 0. The main reason is the L_J is not very large, which is unworthy of adopting FH or PC. When the $L_J > 50$ (i.e., 50 is defined as the loss of FH), the S_T is stabilized around 78%. Here, we consider that if $S_T \geq 75\%$, the anti-jamming scheme is effective because the random jamming rate is set as 25% from the predetermined sweep cycle $1/\lceil K/m \rceil = 0.25$. We also notice that when $15 < L_J \leq 50$, the S_T in the random power mode increases earlier than that in the max power mode, because a certain anti-jamming scheme usually has a better effect on defending against jamming attacks whose signal strength is relatively low.

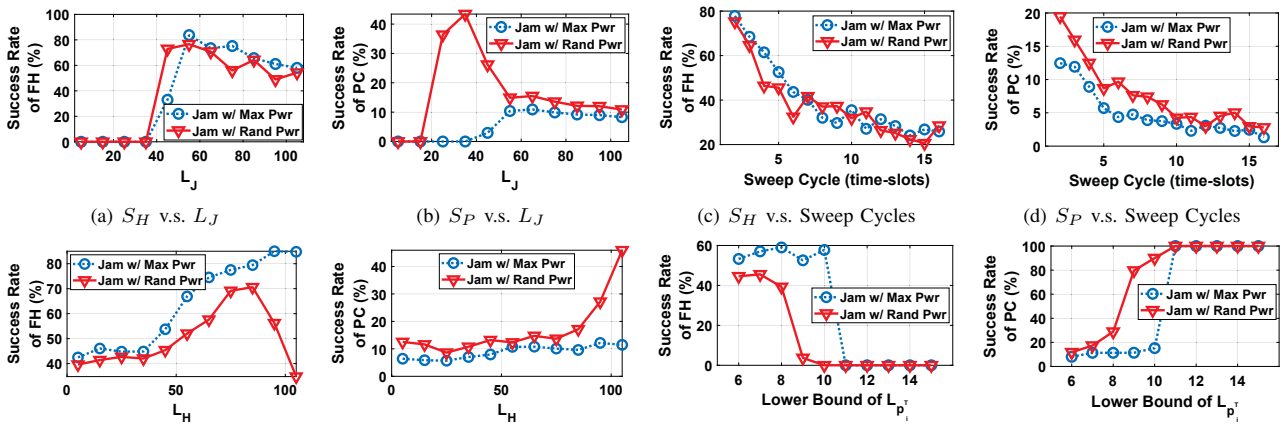
Sweep Cycle. Given a longer sweep cycle, the Tx will less likely to get jammed. Fig. 6(b) shows that the S_T increases with the increase of sweep cycle. Two modes have almost an identical tendency except when $sweep\ cycle = 2$. In this case, the random sweep is degraded into an alternate sweep, which is easier to be predicted.

Impact of L_H. A large L_H will result in the agent being reluctant to adopt FH in mitigating jamming attacks. In Fig. 6(c), the S_T decreases with the increase of L_H. The S_T in the random mode has a significant decrease when $L_H > 85$, because within this range, the loss caused by jamming is close to that caused by FH, the agent does not tend to use FH. However, the S_T in the max mode does not have this feature, because FH is the only effective way to avoid being jammed, which must be adopted.

Impact of $L_{p_i^T}$. We evaluate whether the scope of power level has some impact on strategy selection. Fig. 6(d) shows the variation of S_T with $L_{p_i^T}$. The scope of $L_{p_i}^T$ varies from $[6, 15]$ to $[15, 24]$. When $6 \leq L_{p_i^T} \leq 9$, the S_T in both modes increase slowly. When $L_{p_i^T} \geq 11$, the S_T in both modes maintain at 100%, because the Tx can ensure $SNR > \tau$ within this range.

2) Adoption Rate of FH and PC: Adoption Rate refers to the ratio of the number of time slots adopting FH or PC to the total number of time slots. A good strategy should take actions as few as possible based on the premise of ensuring the success rate of transmission.

Impact of L_J. Fig. 7(a) shows that the A_H increases with the increase of L_J. $L_J = 35$ is an inflection point, before which the A_H maintains at 0, and then, the A_H gradually increases to 50%. Similarly, the variation of A_P with L_J is shown in Fig. 7(b). There are significant differences between the two modes. When $15 < L_J \leq 35$, for the max mode, adopting PC has no effect on improving S_T (because the max power of Jx is always larger than that of Tx). Thus, the agent is less likely



(a) S_H v.s. L_J (b) S_P v.s. L_J (c) S_H v.s. Sweep Cycles (d) S_P v.s. Sweep Cycles

(e) S_H v.s. L_H (f) S_P v.s. L_H (g) S_H v.s. $L_{p_i^T}$ (h) S_P v.s. $L_{p_i^T}$

Fig. 8. Success Rate of FH and PC

to adopt PC. However, for the random mode, PC is adopted extensively, because the cost of PC is less than FH.

Sweep Cycle. Fig. 7(c) and Fig. 7(d) show that both A_H and A_P decrease with the increase of sweep cycle, which indicates that the larger sweep cycle, the less necessary to take anti-jamming actions. From Fig. 7(d), it can be seen that the A_P in the random mode is usually higher than that in the max mode, which also demonstrates that the agent does not intend to adopt PC due to its low effectiveness in the max mode.

Impact of L_H. The increase of L_H will lead to the decrease of A_H, as shown in Fig. 7(e). However, there is an obvious difference between the two modes when $L_H > 85$. In Fig. 7(f), this feature also exists, because FH is the only effective way in the max mode, which must be adopted. For the random mode, it can be abandoned (since PC can undertake the responsibility).

Impact of $L_{p_i^T}$. The increase of $L_{p_i^T}$ will result in the decrease of A_H and the increase of A_P, as shown in Fig. 7(g) and Fig. 7(h). $L_{p_i^T} = 11$ is an inflection point, after which the PC is sufficient to avoid being jammed and FH is no longer necessary.

To sum up, the PC adoption rate is usually higher in the random mode instead of the max mode. However, the FH adoption rate does not have an obvious difference between the two modes. The relatively low PC adoption rate in the max mode can avoid unnecessary and meaningless energy waste, which is of great importance to energy-constrained applications. For users who are concerned about the energy consumption, they can choose a relatively large $L_{p_i^T}$ to train the neural network, so that the PC adoption rate will decrease and the goal of energy conservation will be achieved.

3) Success Rate of FH and PC: It refers to the ratio of the number of successful transmissions using FH or PC to the total number of trials that adopt FH or PC, for which a higher value of S_H or S_P indicates the corresponding actions are useful in anti-jamming.

Impact of L_J. With the increase of L_J, S_H shows a rapid increase (when $35 < L_J < 55$), as shown in Fig. 8(a). This feature indicates that most of the actions are useful. When $L_J \geq 55$, the S_H decreases slowly, because the agent will adopt FH more frequently to avoid being jammed. S_P has a similar feature in Fig. 8(b), except that when $15 < L_J < 55$, two modes are significantly different. This result also verifies that PC is more effective in the random mode.

Sweep Cycle. Fig. 8(c) and 8(d) show that both S_H and S_P decrease with the increase of sweep cycle. FH is the dominant approach that has a high success rate (varies between 77.82% and 20.6%), whereas, for the same sweep cycle, PC has a relatively low success rate (varies between 19.47% and 1.32%).

Impact of L_H. Fig. 8(e) and Fig. 8(f) shows the variation of S_H and S_P with L_H, respectively. A notable feature is that in both figures, when $L_H > 85$, two modes have a marked difference. In the random mode, PC replaces FH as the dominant approach when L_H gets larger, but in the max mode, FH is irreplaceable.

Impact of $L_{p_i^T}$. Fig. 8(g) and Fig. 8(h) show the variation of S_H and S_P with $L_{p_i^T}$, respectively, who have the opposite trend. This trend indicates that PC replaces FH as the dominant approach when $L_{p_i^T}$ is relatively large, because, with the increase of $L_{p_i^T}$, Tx will be able to defeat Jx so that FH is no longer necessary.

To draw a conclusion, the results in Fig. 8 demonstrate that in the case of limited transmission power, FH is more useful than PC and its success rate is also significantly higher than that of PC.

D. Real-World Field Experiments

In the real-world experiment, we need to select a set of proper parameters to train the DQN, and load the training result into the TI CC26X2R1 LaunchPad. We choose $L_J = 100$, $sweep\ cycle = 4$, $L_H = 50$ and $L_{p_i}^T \in [6, 15]$ as the parameters, because these parameters have a better performance according to the simulation results and they are consistent with the WiFi-jamming-ZigBee scenario. After implementing the DQN on the hardware platform, we then (1) evaluate the performance of our anti-jamming scheme in terms of time consumption and goodput; (2) compare its performance with other schemes; and (3) analyze factors that may influence the performance.

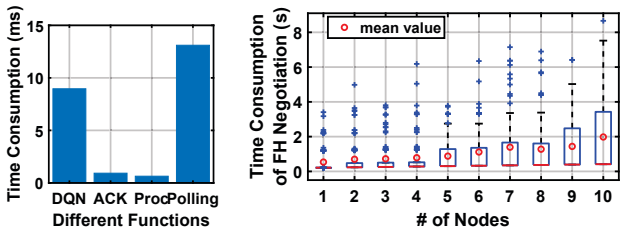
1) Time Consumption: This is a key metric needed to be concerned during the design procedure of the IoT network. It is related to whether multiple nodes can work compatibly with each other. As shown in Fig. 9(a), we measure the time consumption of four typical functions (i.e. the running of the neural network, data transmission, data processing, and FH negotiation). Each function has been tested 100 trials.

Performing DQN. At the beginning of each time slot, the hub needs to use DQN to decide the frequency and power level that should be used, which usually takes 9ms.

Round Trip Time. Sending data and waiting for ACK are the basic task of each peripheral node. A long waiting time will result in a decrease of the data rate. Our result shows that a peripheral node usually has to wait 0.9ms to get the ACK from the hub. Thus, the ACK timeout can be set as a multiple of 0.9ms to ensure the correct receiving of ACK.

Data Processing. After receiving a packet from a peripheral node, the hub also needs some time to process data before it can begin the next round of channel listening. The processing time usually takes 0.6ms. Thus, the overall data rate of peripheral nodes should not be over the limit of 1pkt/0.6ms.

Polling Mode. At the beginning of each time slot, the hub also needs to announce the channel and power level information to each peripheral node, which adopts the polling mode. After ensuring that all peripheral nodes have received the information correctly, the hub will notify them to change frequency (if needed) together. The above procedure usually takes 13.1ms for each node.

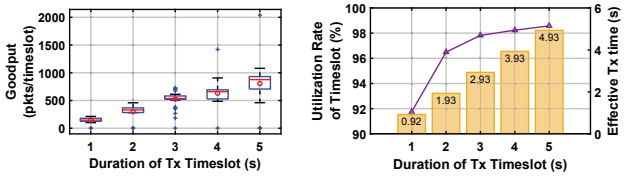


(a) Typical Functions (b) Time Consumption v.s. Network Size

Fig. 9. Time Consumption

Since the polling procedure is significantly time-consuming, we also conduct an experiment to evaluate how much time it will take when the network size is increasing. As shown in Fig. 9(b), the time consumption of negotiation increases with the increase of the number of nodes. In some cases, it can be several seconds, because some nodes may not be in the correct channel, and we need to wait for them to go back to the control channel. The above result can also be used as guidance when designing the size of time slots according to the size of the network.

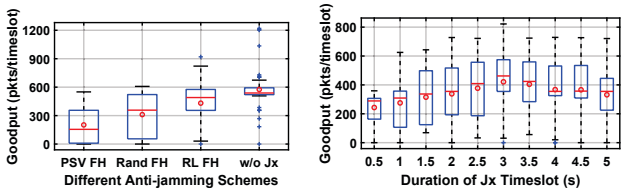
2) Goodput: Goodput [18] refers to the useful information (i.e., payload data instead of ACKs or other control frames) that is delivered to the hub per unit of time. With the increase of time slot duration, the goodput should increase, which has been verified by Fig. 10(a). In particular, the number of received packets per time slot increases from 148 to 806 gradually. Meanwhile, since the IoT network needs to negotiate the FH and PC information per time slot, the proportion of negotiation time will decrease with the increase of time slot duration. This conclusion is verified by Fig. 10(b). It can be seen that the utilization rate of time slots increases from 91.75% to 98.58% gradually. Each time slot, the system still needs to spend about 0.07s for FH negotiation.



(a) Goodput v.s. Timeslot Duration (b) Utilization Rate of Timeslot

Fig. 10. Goodput & Utilization Rate v.s. Timeslot Duration

3) Anti-jamming Effect Comparison: We implement three different anti-jamming schemes (i.e. RL FH, Rand FH, and PSV FH) and compare their efficiency under the same jamming scenario. The implementation of these anti-jamming schemes is not strictly in accordance with any specific literature, we mainly extract the principle features of anti-jamming schemes from some common methods (e.g. [15], [19]). Fig. 11(a) shows the results: (1) Passive FH: Passive FH is a type of common anti-jamming scheme. It adopts FH or PC only when the communication has been jammed. Experiment results show that in our jamming scenario, passive FH can achieve a goodput of 216 packets/time-slot. (2) Rand FH: Rand FH randomly selects FH or PC at the beginning of each time slot. It changes more frequently, thus, it has a higher goodput than passive FH, which is 311 packets/time-slot. (3) RL FH: our scheme has the best performance among these three schemes. It can achieve a goodput of 431 packets/time-slot, which is 2 times/1.39 times that of passive FH/random FH, respectively. In addition, the goodput is 78.5% of that in the normal scenario (i.e., without jamming, whose goodput can reach 575 packets/time-slot). By contrast, passive FH and random FH can only achieve 37.6% and 54.1% of the goodput of the normal scenario, respectively.



(a) Scheme Comparison (b) Influence From Jx

Fig. 11. Schemes Comparison & Analysis

4) Discussion of other practical factors: In addition to the anti-jamming scheme itself, some other factors may also have an impact on the anti-jamming effect, e.g. the relationship between Jx time slot and Tx time slot. If the Jx time slot is shorter than that of Tx, Jx will be able to detect and jam Tx in a quick manner, resulting in the decrease of Tx goodput. If the Jx time slot is longer than that of Tx, the following phenomenon may appear: Jx is staying at and jamming a certain channel while Tx hops back to this channel multiple times, which will also result in the decrease of goodput. To verify this

conclusion, we set the Tx time slot as 3s and vary the Jx time slot (from 0.5s to 5s) to see the variation of goodput. Fig. 11(b) shows the result. It can be seen that when the duration of the Jx time slot is also 3s, the anti-jamming scheme has the best performance, whose goodput can reach 421 packets/time-slot. However, when Jx has a larger or smaller time slot duration, the performance of the anti-jamming scheme will degrade. Due to the limitation of hardware, the time slot cannot be too small (i.e., <0.5s), otherwise, FH negotiation will occupy the whole time slot and there is no sufficient time to transmit data.

Based on our experimental results, our scheme can successfully defend against the jamming attack and outperform existing schemes. It can reach 2 times / 1.39 times more than the goodput of passive and random anti-jamming designs.

V. Related Works

A. Cross-Technology Jamming

JamCloak [13] propose a reactive jamming attack over CTC links. They extract essential features to classify the CTC traffic and then, design jamming signals that can effectively attack the specific CTC protocol. SamBee [20] implement a parallel spoofing system that can spoof the ZigBee devices operating in two different channels or jam the ZigBee devices operating in five distinct channels simultaneously only using a single WiFi frame. In [11], the author proposes a new attack that an adversary pretends to be a legitimate WiFi device and sends out WiFi packets to prevent ZigBee devices' communication or colliding with ZigBee's packets. DeepJam [14] proposes a stealthy jamming strategy to jam ZigBee traffic. It relies on deep learning techniques to capture the temporal pattern of the past wireless traffic and predict the future wireless traffic. Wi-attack [21] proposes an impersonation attack that uses WiFi devices to attack iBeacon services, which can bring an average distance error of up to 20 meters in a common fingerprint-based localization system. In [12], [22], [23], the authors propose several cross-technology attacking and defensive approaches to deal with the scenario that a WiFi device overhear and emulate the ZigBee waveform to attack ZigBee IoT devices.

B. Anti-jamming Techniques

Various techniques have been proposed to defend against jamming attack, such as: channel hopping [24]–[26], spreading spectrum [27], [28], MIMO-based technique [29], [30], coding technique [31], power control [32], [33], reinforcement learning-based technique [34], [35], or a combination of multiple methods [15], [36].

Tri-CH [24] is a channel hopping algorithm for CRNs, which adopts a random jump pattern to achieve a high security level and stay pattern to guarantee bounded time to rendezvous. RD-DSSS [27] is a Randomized Differential DSSS (RD-DSSS) scheme to achieve anti-jamming broadcast communication without shared keys. It encodes data using the correlation of unpredictable spreading codes. In [29], Yan et al. present a MIMO-based anti-jamming scheme that exploits interference cancellation and transmit precoding capabilities of MIMO technology. In [31], Yue et al. present two coding schemes for recovering lost packets transmitted through parallel channels. In [15], [36], Hanawal et al. propose a scheme to mitigate jamming by jointly optimizing the frequency hopping and rate adaptation techniques.

VI. Conclusion

In this paper, we demonstrate a new type of cross-technology jamming attack in a heterogeneous IoT system, which has a better attacking effect and a stronger stealthiness. To deal with the jamming attack, we propose an anti-jamming approach that jointly uses frequency hopping and power control techniques. The optimal strategy of choosing channel and power level is acquired by applying deep Q-learning. Extensive experiments show that our approach can achieve 2 times/1.39 times the goodput of passive FH/random FH, respectively. In addition, its goodput can achieve 78% that of normal scenario (i.e. no jammer).

References

[1] S. R. Department. Number of internet of things (iot) connected devices worldwide in 2018, 2025 and 2030. [Online]. Available: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[2] T. Pulkkinen, J. K. Nurminen, and P. Nurmi, "Understanding wifi cross-technology interference detection in the real world," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 954–964.

[3] Y. Hou, M. Li, X. Yuan, Y. T. Hou, and W. Lou, "Cooperative cross-technology interference mitigation for heterogeneous multi-hop networks," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 880–888.

[4] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 2–14.

[5] X. Guo, Y. He, J. Zhang, and H. Jiang, "Wide: physical-level ctc via digital emulation," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2019, pp. 49–60.

[6] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "Iot goes nuclear: Creating a zigbee chain reaction," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 195–212.

[7] X. Cao, D. M. Shila, Y. Cheng, Z. Yang, Y. Zhou, and J. Chen, "Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 816–829, 2016.

[8] A. Burg, A. Chattopadhyay, and K.-Y. Lam, "Wireless communication and security issues for cyber–physical systems and the internet-of-things," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 38–60, 2017.

[9] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3453–3495, 2018.

[10] X. Zhang, P. Huang, L. Guo, and Y. Fang, "Hide and seek: Waveform emulation attack and defense in cross-technology communication," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 1117–1126.

[11] Z. Chi, Y. Li, X. Liu, W. Wang, Y. Yao, T. Zhu, and Y. Zhang, "Countering cross-technology jamming attack," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 99–110.

[12] S. Yu, X. Zhang, P. Huang, L. Guo, L. Cheng, and K. Wang, "Authctc: Defending against waveform emulation attack in heterogeneous iot environments," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 20–32.

[13] G. Chen and W. Dong, "Jamcloak: Reactive jamming attack over cross-technology communication links," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 2018, pp. 34–43.

[14] D. Han, A. Li, L. Zhang, Y. Zhang, J. Li, T. Li, T. Zhu, and Y. Zhang, "Deep learning-guided jamming for cross-technology wireless networks: Attack and defense," *IEEE/ACM Transactions on Networking*, 2021.
[15] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2247–2259, 2015.
[16] S. Yu, X. Zhang, P. Huang, and L. Guo, "Physical-level parallel inclusive communication for heterogeneous iot devices," in *the 41st IEEE International Conference on Computer Communication (IEEE INFOCOM)*. IEEE, 2022, pp. 1–10.
[17] R. Bellman, "Dynamic programming, isbn 0486428095 ed," 1957.
[18] Wikipedia. Goodput. [Online]. Available: https://en.wikipedia.org/wiki/Goodput
[19] G.-Y. Chang, J.-F. Huang, and Z.-H. Wu, "A frequency hopping algorithm against jamming attacks under asynchronous environments," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 324–329.
[20] D. Gao, S. Wang, Y. Liu, W. Jiang, Z. Li, and T. He, "Spoofing-jamming attack based on cross-technology communication for wireless networks," *Computer Communications*, 2021.
[21] X. Na, X. Guo, Y. He, and R. Xi, "Wi-attack: Cross-technology impersonation attack against ibeacon services," in *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2021, pp. 1–9.
[22] S. Yu, X. Zhang, P. Huang, and L. Guo, "Secure authentication in cross-technology communication for heterogeneous iot," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2019, pp. 1–2.
[23] X. Zhang, S. Yu, H. Zhou, P. Huang, L. Guo, and M. Li, "Signal emulation attack and defense for smart home iot," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, pp. 1–15, 2022.
[24] G.-Y. Chang, S.-Y. Wang, and Y.-X. Liu, "A jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 10, pp. 6712–6725, 2017.
[25] Y. Shi, K. An, and Y. Li, "Index modulation based frequency hopping: Anti-jamming design and analysis," *IEEE Transactions on Vehicular Technology*, 2021.
[26] Y. Gao, Y. Xiao, M. Wu, M. Xiao, and J. Shao, "Game theory-based anti-jamming strategies for frequency hopping wireless communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5314–5326, 2018.
[27] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized differential dsss: Jamming-resistant wireless broadcast communication," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.
[28] C. Popper, M. Strasser, and S. Capkun, "Anti-jamming broadcast communication using uncoordinated spread spectrum techniques," *ieee journal on selected areas in communications*, vol. 28, no. 5, pp. 703–715, 2010.
[29] Q. Yan, H. Zeng, T. Jiang, M. Li, W. Lou, and Y. T. Hou, "Mimo-based jamming resilient communication in wireless networks," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 2697–2706.
[30] W. Shen, P. Ning, X. He, H. Dai, and Y. Liu, "Mcr decoding: A mimo approach for defending against wireless jamming attacks," in *2014 IEEE Conference on Communications and Network Security*. IEEE, 2014, pp. 133–138.
[31] G. Yue and X. Wang, "Anti-jamming coding techniques with application to cognitive radio," *IEEE transactions on wireless communications*, vol. 8, no. 12, pp. 5996–6007, 2009.
[32] S. Lv, L. Xiao, Q. Hu, X. Wang, C. Hu, and L. Sun, "Anti-jamming power control game in unmanned aerial vehicle networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.
[33] Y. Chen, Y. Li, D. Xu, and L. Xiao, "Dqn-based power control for iot transmission against jamming," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 2018, pp. 1–5.
[34] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H. V. Poor, "Two-dimensional antijamming mobile communication based on reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9499–9512, 2018.
[35] Y. Bi, Y. Wu, and C. Hua, "Deep reinforcement learning based multi-user anti-jamming strategy," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.
[36] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Game theoretic anti-jamming dynamic frequency hopping and rate adaptation in wireless systems," in *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2014, pp. 247–254.

APPENDIX

Proof of Theorem III.1

Proof. To prove this proposition, we need to use the Banach's fixed point theorem and some related concepts, as shown in theorem A.1, definition 1 and 2.

Theorem A.1. *(Banach's Fixed Point Theorem) Let (M, d) be a complete metric space, and f be a contraction mapping on M. Then, there exists a unique fixed-point $x^* \in M$ such that $f(x^*) = x^*$. Furthermore the x^* can be found as follows: $x^* = f(f(f(...f(x)...)))$.*

Definition 1. *(Metric Space) A metric space is an ordered pair (M, d) where M is a set and d is a metric on M, i.e., a function $d : M \times M \to \mathbb{R}$ such that for any $x, y, z \in M$, the following holds: (1) $d(x, y) = 0 \Leftrightarrow x = y$; (2) $d(x, y) = d(y, x)$; (3) $d(x, z) \leq d(x, y) + d(y, z)$.*

Definition 2. *(Contraction Mapping) A function f defined on the metric space (M, d) is a contraction mapping if there exists some constant $\gamma \in [0, 1)$ such that for any two elements $x, y \in M$, $d(f(x), f(y)) \leq \gamma d(x, y)$.*

If we can show that the Bellman optimality equation $V_*(x)$ is a contraction mapping for some metric space (M, d), then, by the Banach fixed point theorem, we can conclude that the repeated application of the Bellman optimality equation will eventually give a unique optimal state-value function, using which the optimal policy can be derived.

We use the L-infinity norm as the metric shown below,

$$\|X\|_\infty = \max_i |X_i| \tag{28}$$

according to which, the distance between the two elements is equal to the highest element-wise absolute difference between the two. Since the results of repeatedly applying $V(s)$ always stay in the real space, the metric space is complete.

Then, we can prove that the $V(s)$ is a contraction mapping in the metric space (X, l_∞) as follows:

$$\begin{aligned}
&\|V_1(x) - V_2(x)\| \\
&= \left\| \max_a \left(U(x,a) + \gamma \sum\nolimits_{x'} P(x'|x,a) V_1(x') \right) \right. \\
&\quad \left. - \max_{a^*} \left(U(x,a^*) + \gamma \sum\nolimits_{x'} P(x'|x,a^*) V_2(x') \right) \right\| \\
&\leq \left\| \max_a \left(U(x,a) + \gamma \sum\nolimits_{x'} P(x'|x,a) V_1(x') \right) \right. \\
&\quad \left. - \max_a \left(U(x,a) + \gamma \sum\nolimits_{x'} P(x'|x,a) V_2(x') \right) \right\| \\
&= \left\| \max_a \left(\gamma \sum\nolimits_{x'} P(x'|x,a) (V_1(x') - V_2(x')) \right) \right\| \\
&\leq \gamma \max_a \sum\nolimits_{x'} P(x'|x,a) \|V_1(x') - V_2(x')\| \\
&= \gamma \|V_1(x') - V_2(x')\|
\end{aligned} \tag{29}$$

Since (X, l_∞) is a complete metric space, and $V(x)$ is a contraction mapping, we conclude that there exists a unique optimal state-value function $V_*(x)$ for every MDP. □