Mitigating Cross-Technology Interference in Heterogeneous Wireless Networks based on Deep Learning

Weidong Zheng, Junmei Yao and Kaishun Wu
{zhengweidong2018, yaojunmei, wu}@email.szu.edu.cn
Shenzhen University, Shenzhen, China,

Abstract—With the prosperity of Internet of Things, a large number of heterogeneous wireless devices share the same unlicensed spectrum, leading to severe cross-technology interference (CTI). Especially, the transmission power asymmetry of heterogeneous devices will further deteriorate this problem, making the low-power devices prohibited from data transmission and starved. This paper proposes an enhanced CCA (E-CCA) mechanism to mitigate CTI, so as to improve the performance and fairness among heterogeneous networks. E-CCA contains a signal identification design based on deep learning to identify the signal type within a tolerable time duration, it also contains a CCA adaptive mechanism based on the signal type to avoid CTI. As a result, the ZigBee devices could compete for the channel with WiFi devices more fairly, and the network performance can be improved accordingly. We set up a testbed based on TelosB, a commercial ZigBee platform, and USRP N210, which can be used as the WiFi platform. With the collected signals through USRP N210, over 99.9% signal identification accuracy can be achieved even when the signal duration is tens of microseconds. Simulation results based on NS-3 shows that E-CCA can increase the ZigBee performance dramatically with little throughput degradation for WiFi.

Keywords-ZigBee; WiFi; Signal identification; Cross-technology interference; Heterogeneous wireless networks

I. Introduction

The number of Internet of Things (IoT) devices is anticipated to reach 25.1 billion by 2021 [1]. The IoT era is coming. This boosts proliferation of heterogeneous wireless technologies sharing the same unlicensed spectrum, such as WiFi [2], ZigBee [3] and Bluetooth at the ISM (industrial, scientific and medical) 2.4 GHz band, leading to severe cross-technology interference (CTI).

Wireless devices operating in the ISM bands typically utilize the CSMA/CA (carrier sense multiple access with collision avoidance) mechanism to avoid interference. Before transmitting, the device listens to the channel and samples the energy on the channel to determine the channel availability: if the energy is over a predefined CCA (clear channel assessment) threshold, the channel is busy; otherwise, it is idle and the device can proceed with the transmission. Due to heterogeneous application requirements, the IoT devices transmit signals at different power levels. The low-power devices (such as ZigBee) always transmit signals at less than 1 mW for energy reservation, while the high-power devices (such as WiFi) can transmit signals up to 100 mW for larger coverage. Since the CCA threshold is fixed in each device, this power asymmetry further deteriorates the CTI problem, as low-power devices are always prohibited from transmission by high-power devices.

Researchers have paid much attention to this problem, and put forward several kinds of solutions. Some works make devices switch the channel to avoid spectrum overlap [4], this becomes less effective since the number of devices increases dramatically, and the devices are hard to find a vacant channel for use. Some works propose subcarrier nullification in WiFi to reserve subchannels for ZigBee, so as to avoid CTI [5]; these methods require hardware modifications on both the transmitter and receiver, making it hard to be deployed. Some recent works utilize cross-technology communication to negotiate the channel usage [6], these works are still less effective under the power asymmetry scenario.

In this paper, we present an enhanced CCA (E-CCA) mechanism, which adjusts the CCA threshold adaptively based on signal identification through deep learning to mitigate CTI in heterogeneous wireless networks, so as to improve the overall network performance. This design is based on an observation on power asymmetry: a WiFi device can still transmit a packet and corrupt the ongoing ZigBee transmissions, because the energy of the ZigBee signal perceived by WiFi is too weak to achieve the CCA threshold and the channel is considered to be idle. Our experiments reveal that the hidden pattern of the weak ZigBee signal can be extracted from the raw IQ data received by WiFi through deep learning. Thus, E-CCA contains a signal identification design based on deep learning to make the signal type identified within a tolerable time duration, it also contains a CCA adaptive mechanism based on the signal type to avoid CTI. As a result, the ZigBee devices could compete for the channel with WiFi devices more fairly, and the network performance can be improved accordingly.

This paper makes the following main contributions:

- We design E-CCA, a CCA adaptive mechanism based on signal identification through deep learning to mitigate CTI in heterogeneous wireless networks.

2690-5965/20/$31.00 ©2020 IEEE
DOI 10.1109/ICPADS51040.2020.00040

- To the best of our knowledge, it is the first time that deep learning is utilized to identify ZigBee and WiFi signals transmitted by commercial platforms and sampled by USRP N210. Over 99.9% signal classification accuracy can be achieved even when the signal duration is tens of microseconds.
- We evaluate the performance of E-CCA through simulations on NS-3. The results indicate that the performance of ZigBee can be largely increased, with little throughput sacrifice to WiFi.

This paper is organized as follows: Section II introduces recent works related to this paper. Section III briefly describes the fundamental knowledge and main difference of WiFi and ZigBee in PHY and MAC, then describes how cross-technology interference occurs. Section IV provides a basic diagram to show our design. Section V presents the procedure of signal identification and its performance. Finally, the network performance evaluation would be shown in Section VI.

II. Related Works

In this section, we will discuss existing solutions related to this work.

A. Signal Identification

Effectively recognizing other signals is the first step of cross-technology interference elimination. We see a lot of research works focusing on this area in recent years. Traditional method is to analyze the signal through a dedicated spectrum analyzer, which could achieve high accuracy due to the high sampling rate. However, it is infeasible to be deployed in off-the-shelf devices. Many researches intend to find out obvious features to recognize different kinds of signals. ZiFi detects the existence of WiFi on ZigBee devices through the periodicity of WiFi Beacon [7]. Airshark exploits RSSI sampled by commodity WiFi cards to classify multiple protocols [8]. Most works extract features related to RSSI and CSI to build different classifier [9]–[11]. But human extracted features are no longer feasible when ZigBee signal is weak, especially within a short duration. So, deep learning, which could automatically extract features from raw data, is taken into consideration.

Deep Learning shines in computer vision and speech signal recognition. LeNet is the earliest CNN and performs well in handwritten digit recognition [12]. AlexNet receives a lot of attention while achieving 84.7% accuracy in ImageNet dataset [13]. Then, many efforts enhance the power of deep learning in classification tasks. Deep learning based method could utilize raw signal samples under low SNR to recognize different modulation [14]. This motivates us to identify ZigBee signals from raw samples captured by WiFi AP.

B. Cross-Technology Interference Mitigation

The cross-technology interference in heterogeneous networks is a tough obstacle to be solved for a long time. There are many works evaluating the performance study of WiFi impact on ZigBee link with the consideration of various distances and different transmission power [15]. The WiFi network can usually cause about 60% packet loss rate to the ZigBee network, which can reach 87% in extreme cases. Therefore, it is urgent to find a solution to reduce the interference of WiFi to ZigBee.

It is straightforward to switch to another idle channel without signal for communication. RSSI-Based interference detection mechanism is proposed to search for a free channel [16]. However, spectrum resource is limited and valuable. With the prosperity of IoT, it is infeasible to find a free channel. Cooperation would be imperative.

Some researchers solve the problem by increasing ZigBee visibility. Cooperative Busy Tone mechanism via placing a cooperative ZigBee node to send dummy messages besides WiFi transmitter is proposed so that the channel would be reserved [17]. Weeble changes the Length field in WiFi preamble to control WiFi transmission [18]. Both methods require another specific node, which is indirect and costly. Some choose to improve robustness. Liang et al. turn to using dummy header and more robust channel coding like RS code to improve success rate in decoding [19]. WiCop adopted similar idea that increases ZigBee preamble so that ZigBee transmission could be detected by WiFi [20].

Some recent researchers intend to solve this problem through cross-technology communication (CTC), which establishes direct message exchange between heterogeneous devices. For example, WEBee modifies the payload in WiFi packet to emulate a ZigBee signal and enable communication from WiFi to ZigBee [21]. The capability of CTC brings great opportunities to perform cross-technology coordination mechanisms. ECT utilizes CTC to let a server schedule ZigBee transmissions [22]. CTC is a promising way to solve this problem, but requires some physical layer modifications.

Subcarrier nullification reserves some subcarriers and creates a notch in its spectrum to accommodate narrowband signals [5], [23]. However, this technology requires modification on both the transmitter and receiver. Our design only needs simple modification on the transmitter, which is more feasible for deployment.

III. Background

Here we first introduce the PHY and MAC specifications for WiFi and ZigBee, then illustrate how cross-technology interference affects their performance.

A. The PHY Layer Specifications

1) Channel Allocation:

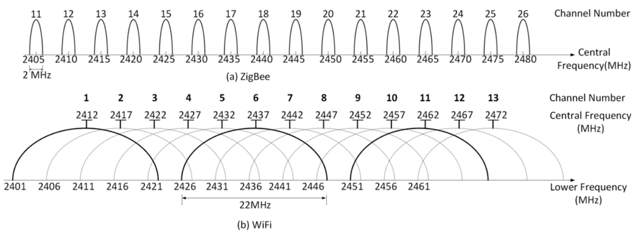


Fig 1. Overlap of WiFi and ZigBee channels.

Each ZigBee channel occupies 2 MHz, its central frequency is calculated as follows:

$$F_{c_{zigbee}} = 2405 + 5(k - 11)MHz, k = 11, 12, \ldots, 26$$

Each WiFi channel bandwidth is 20 MHz, its central frequency is calculated as follows:

$$F_{c_{wifi}} = 2412 + 5(k - 1)MHz, k = 1, 2, \ldots, 13$$

Fig. 1 illustrates how the WiFi channels overlap with ZigBee channels. Each WiFi channel would cover up to four ZigBee channels. Due to the limited spectrum resources, the channel overlapping situation is unavoidable because each ZigBee channel would be covered by multiple WiFi channels.

2) The PHY Parameters:

Table I illustrates the difference of physical layer demand between ZigBee and WiFi. Nowadays, the majority of ZigBee devices operate with transmit powers at about 0 dBm to cut down energy consumption, while most WiFi devices transmit at 16 dBm with the purpose of covering more area. Due to the difference in throughput demand, they adopted different modulation methods, which is the barrier of channel allocation negotiation.

B. The MAC Layer Mechanism

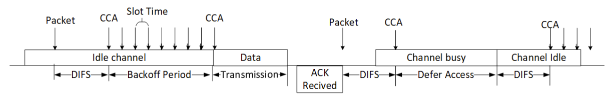
Both the WiFi and ZigBee networks adopt Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) mechanism to access the channel, Fig. 2 shows how CSMA/CA works.

CSMA/CA conducts Clear Channel Assessment (CCA) before data transmission to check if the channel is idle. CCA performs mainly by energy detection, which judges the channel as idle if the signal energy is less than a predefined threshold.

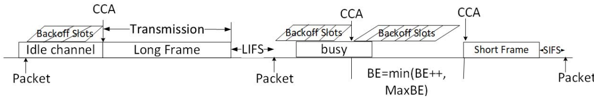
The CSMA/CA mechanism of WiFi is shown in Fig. 2(a). If the transmitter intends to send a packet, it would wait for Distributed Interframe Space (DIFS). If the channel is considered idle during the DIFS time, the transmitter still waits for a random duration (contention window) which consists of multiple backoff slots. It performs CCA in each slot to detect the channel: the backoff counter will decrease if the channel is idle, and hang up once the channel is busy. Transmission will start when the backoff counter reaches zero. Once the transmission fails, the device will defer until the channel is idle and use Binary Exponential Backoff (BEB) mechanism to contend the channel for retransmission. With several failed retransmissions, the packet will be dropped.

Table I: Physical Parameters Comparison

IEEE Standard	802.15.4	802.11n
Modulation Method	OQPSK	OFDM
Tranmission Rate	250 kbps	300 Mbps
Tranmission Power	-3 ~ 6 dBm	12 ~ 20 dBm
Transmission Range	10 ~ 30 m	30 ~ 100 m



(a) WiFi



(b) ZigBee

Fig 2. CSMA/CA Mechanism

The CSMA/CA mechanism adopted by ZigBee shown in Fig. 2(b) is slightly different. Every time the device intends to transmit, it would select a random unit from 0 to $2^{BE} - 1$ to delay before performing CCA, where the backoff exponent (BE) is used to determine the number of backoff slots. If the CCA result is idle, the transmission start; otherwise, BE increases by one until BE_{max} and try backoff delay again. After three retransmissions, the packet would be dropped.

The difference in MAC settings is that, the WiFi slot is 9 us while the ZigBee slot is 320us, which makes WiFi have more chances to access the channel. Meanwhile, WiFi performs CCA in every slot, but ZigBee performs only once after backoff to reduce power consumption.

C. Cross-Technology Interference

As mentioned above, with 16 dB higher transmission power, the coverage area of WiFi is 3x even 10x than ZigBee. Under this power asymmetric scenario, WiFi could trigger ZigBee CSMA/CA backoff but not vice versa. As shown in Fig. 3, When the ZigBee device Z1 is transmitting data to another device Z2, the transmission may not be sensed by WiFi AP as its transmission power is too low to achieve the CCA threshold. Thus, AP would consider the channel as idle and start transmitting a data packet, which would induce interference at Z2, leading to the ZigBee packet corruption. The WiFi packet may also be slighted interfered at STA, but it could be recovered through forward error correction due to the robustness of wide-band signal. As a result, WiFi transmission succeeds but the ZigBee transmission fails.

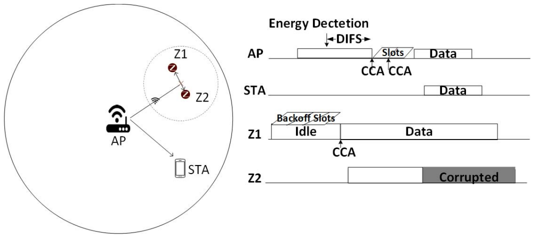


Fig 3. A Scenario of Cross-Technology Interference

We have carried out simulations based on this assumption, and found out that the ZigBee packet receive rate (PRR) would be lower to 20%, even approximate to zero in some extreme cases.

IV. System design

In this paper, we propose E-CCA to combat the cross-technology interference under the coexistence of WiFi and ZigBee networks. As shown in Fig. 4, E-CCA consists of two parts, one is the signal identification mechanism aiming to detect weak ZigBee links via deep learning, the other is the MAC protocol design for improving the performance and fairness in the heterogeneous networks.

To be concrete, let's see the CSMA/CA mechanism in Fig. 2. Before transmitting, the WiFi AP obtains some sampled signal and performs CSMA/CA to determine the channel state. When a ZigBee device is transmitting, the sampled signal would be too weak to achieve the CCA threshold. The AP would initiate transmission and interfere with ZigBee's signal reception. However, with the E-CCA design shown in Fig. 5, these sampled signals would first be sent into the deep learning classifier. Via deep learning, the AP could extract hidden patterns under low SNR of the ZigBee signal to tell the existence of ZigBee links in a short DIFS time. Once the ZigBee link is detected, a ZigBeeOn flag will be set true and then signaled to the MAC layer. With this flag, the AP MAC would lower the CCA threshold, consider the channel to be busy, and defer its transmission to reserve the channel for ZigBee.

With E-CCA design, the situation of Fig. 3 changes. When a WiFi device wants to transmit a packet during the ZigBee transmission, it would first perform both energy detection and deep learning during DIFS. It could identify the weak ZigBee signal through deep learning, adjust the CCA threshold in the MAC layer, and consider the channel as busy according to the detected energy. Then the WiFi device would defer its transmission for a short time and start to contend the channel again, while the ZigBee device can finish its transmission without corruption during this reserved time. Therefore, these two kinds of devices could access the channel more fairly.

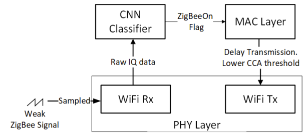


Fig 4. E-CCA Overview

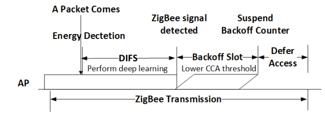


Fig 5. E-CCA MAC Layer

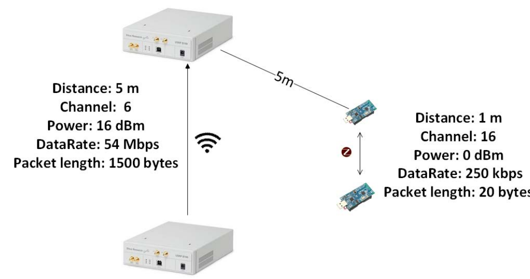


Fig 6. The Testbed Topology

V. Signal Identification

In this section, we first describe the testbed for collecting data and the labeling method, then build a convolutional neural network (CNN) model for performance test.

A. Testbed setup

To sample the wireless signal, we utilize universal software radio peripheral (USRP) N210 as the WiFi transceiver. Driven by GNU Radio in Ubuntu OS, USRP could sample the signal around and analyze. One USRP emits the standard WiFi signal while another USRP operates nearby to collect data. Meanwhile, two commercial ZigBee platforms, TelosB motes, communicate with each other following the standard ZigBee protocol. The testbed topology is shown in Fig. 6. Parameters at the transmitter side are modified as follows.

- Central Frequency: Channel Overlap is unavoidable. In the experiment, we choose the WiFi channel as 6 and the ZigBee channel as 16, which would lead to severe performance degradation to ZigBee
- Transmission power: In this experiment, we choose 16dBm as the default WiFi transmission power, while make TelosB motes transmit at 0dBm.
- Transmission rate: The WiFi transmitter keeps flooding in 54 Mbps to send packets with the length of 1500 bytes, while TelosB Tx mote sends 20 bytes each 1 ms.

To eliminate the interference from other signals, we conduct the data samples in a shielding chamber. We collected different types of signals for training and testing. The data stream we obtained consists of pure ZigBee signals and WiFi signals for training. We also gather mixed ZigBee-WiFi signals samples to build the testing dataset, in the hope of testing the robustness of signal classification.

After acquiring data from USRP, we run offline deep learning training via keras API to build classifier on Dell Latitude 3490 with Intel Graphics 620 and Intel Core i7-8550U CPU.

B. Labeling task

1) Labeling under high resolution: It is easy to obtain label for data with longer sample length, which has high resolution. If we choose 1024 samples to analysis, the most obvious features are the central frequency and bandwidth in the frequency domain. In this situation, Fast Fourier Transform (FFT) can be performed to extract the spectral features. To reduce the ripper and smooth the profile of the waveform, we use hanning window to minimize the signal side lobe. From the spectrum, we could find peaks via simple comparison of neighboring values. The central frequency lies in the middle of peaks, and the width of peaks determines the signal bandwidth.

The spectral features mentioned above could easily tell the difference of the WiFi and ZigBee signals, and further identify the ZigBee signal. The spectral features are obvious and could be easily extracted with 1024 samples. So, we could easily tag the signal type while we choose the FFT bins as 1024.

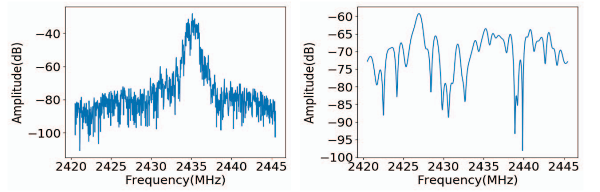
2) Resolution-based feature limitation: The spectral features are limited by the resolution. In other words, with fewer IQ raw data used in FFT, the bandwidth would be obscurer. But it is crucial to use less samples for rapid transmission decision in our design.

As shown in Fig. 7, with 1024 FFT bins, the features are easily distinguishable and the label can be exact. However, with 64 bins, it would be hard to manually extract the features correctly.

As we all known, machine learning lies highly upon features. The features we use determine the performance of the machine learning model. However, when the feature is obscure as the FFT bins are smaller, machine learning based signal classification would be no longer feasible.

Deep learning is data-driven. If tremendous data with corresponding accurate labels are fed into the deep learning model, it can automatically find out features of data and then identify the type of this signal. The advantage of automatically feature extraction is to find hidden patterns that cannot be discovered manually so far.

3) Labeling with fewer samples: The challenge we face is to tag the label for data with shorter sample length. We adopt an indirect way to do it. We first tag the label for data with long sample length, then partition it into shorter samples. In this case, we could believe that the label corresponding to longer samples would still be



(a) bins=1024 (b) bins=64

Fig 7. FFT results with different FFT bins

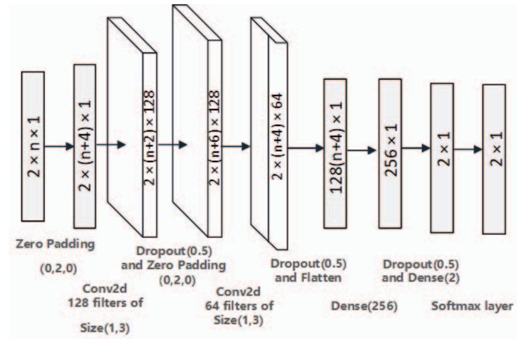


Fig 8. The CNN Structure

suitable for the partitioned shorter samples. For example, since the data of 1024 samples has distinguishable spectral features, it can be labeled easily. When each 1024-sample data is partitioned into several 64-sample data, the label corresponding to 1024 samples would be the same as 64 samples. However, the boundaries of the 1024 samples may be pure noise, leading to inaccurate label in the shorter samples. To avoid confusion, we remove the two boundaries to ensure the label accuracy.

C. Data Transformation

The raw IQ data sampled from USRP is stored as a float32 data stream. To be used by deep learning model, the data stream should first be transformed as a 4-dimensional vector with numpy, and then serialized and stored by cPickle. The 4-dimensional data vector has the form of $N_{examples} \times Dim_{IQ} \times Dim_{value} \times Dim_{channel}$, where $N_{examples}$ represents the amount of examples, $Dim_{IQ} = 2$ represents the I and Q channels, Dim_{value} is the FFT size, and $Dim_{channel} = 1$ is a commonly used value in imagery. For example, If we have 20,480,000 IQ data stream sampled from USRP, and we choose FFT bins as 1024, then the data vector is $10,000 \times 2 \times 1024 \times 1$.

D. CNN Model Structure

The input of the CNN model is the packaged vector, and the output is the judgement of ZigBee signals. The network structure we use is similar to VT-CNN2 [24], the details are shown in Fig. 8. The sample length n illustrated in Fig. 8 varies from 64 to 1024.

Firstly, the samples fed in will go through two convolution layers. These two layers consist of 128 and 64 filters respectively, with size (1,3) and stride (1,1). The activation function used to perform non-linear transformation

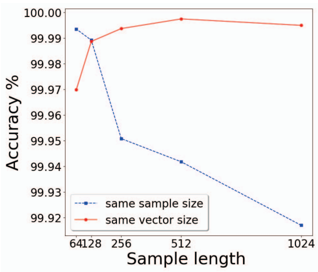


Fig 9. Testing accuracy with sample length

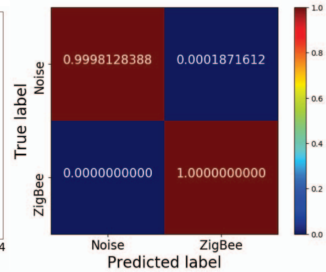


Fig 10. Confusion matrix on sample length 64

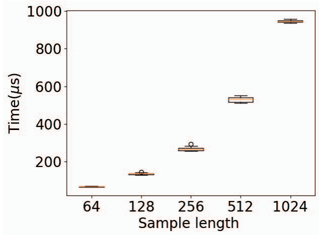


Fig 11. Average testing time of different sample length

is rectified linear (ReLU), which could converge faster. Every time after convolution layer, we would randomly drop out half of the neurons to avoid overfitting and improve the model robustness. The multi-dimensional data output from the second convolution layer will be sent into the flatten layer to be reshaped as a one-dimensional vector, which will be input of the following dense layer. After that, we use softmax function on the one-hot output layer. This model usually converges after about 10 epochs to generate the corresponding result.

E. Identification Accuracy

The training dataset consists of 99,174×2×1024 ZigBee samples and 100,000×2×1024 WiFi samples. The ZigBee samples in the testing dataset occupy about 7% in total, the other are considered as noise. The metric we intend to testify is the classification accuracy. If we divided the samples length into 64, the datasets would be 1586784×2×64 ZigBee samples and 16000000×2×64 WiFi samples. That means the vector size of ZigBee samples increases from 99,174 to 1586784 if the sample length changes from 1024 to 64. When the sample length decreases, the total vector size fed into the classifier increases, making the accuracy increases finally.

In the experiment, we testify the accuracy under the same vector size and the same sample size with different sample length. The accuracy of both situations achieves more than 99%, as shown in Fig. 9. From this figure, it is not surprising that with the same training vectors, longer sample length leads to higher accuracy in classifying the ZigBee signal. However, if we use the same collected samples, which makes the shorter sample length with larger vector size, resulting in higher accuracy.

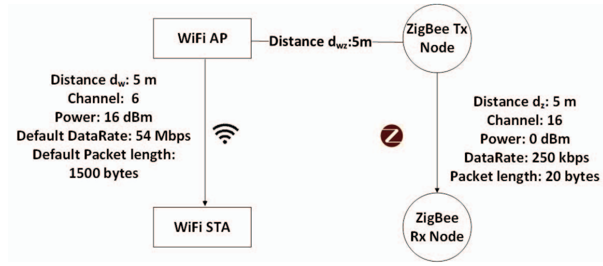


Fig 12. Simulation Topology

The confusion matrix is shown as Fig. 10. The true positive rate of the ZigBee signal is 100%, which means all ZigBee signals can be identified correctly. The false positive rate equals to 0.0187%, which means that a little portion of WiFi signals are classified as ZigBee, this false positive result would lead to the degradation of WiFi throughput. Fortunately, the false positive rate is so small that it would not cause too much influence.

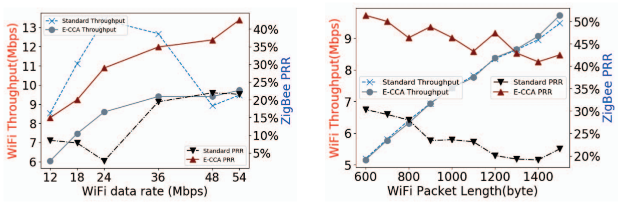
Another advantage of deep learning is that, although the training process may last for a long time, the test time is rather short. We have evaluated the time response of the prediction process. Specifically, we have got the total running time of 10000 inputs, then divided the time by 10000 to get the average time response for each input. To avoid randomness, each experiment is conducted 10 times for statistical convergence. As shown in Fig. 11, when the sample length is doubled, the testing time is doubled too. It would last for about 66us on average when the sample length is 64. Hopefully, with dedicated AI chips such as Tensor Processing Unit (TPU) which are about 15X – 30X faster than GPU or CPU [25], the time could be reduced to about 7 us. In this situation, the signal identification process can be completed within the DIFS time.

VI. Simulation Result

Due to the limitation of computing deep learning in USRP, we turn to simulation to evaluate the network performance. NS-3 is a discrete-event network simulator for Internet systems, its powerful classes could be used to simulate the real-world environment. We note that E-CCA can be easily deployed to commercial WiFi APs when Neural Network Processing Unit (NPU) is added.

We conduct the simulations under a scenario with two links, one adopts ZigBee (802.15.4) protocol and the other adopts WiFi (802.11). The default topology and default transmission parameters are shown in Fig. 12. We would alter the WiFi transmit parameters such as packet length and data rate to see how our design fit under different WiFi situations. We would also change the distance between the devices to build another scenario. Each simulation lasts for 10 seconds.

Due to the demand of high speed, we evaluate the performance of WiFi through the metric of throughput. Meanwhile, we use packet receive rate (PRR) as the



(a) DataRate (b) PacketLength

Fig 13. Performance under different WiFi Setting

metric to evaluate ZigBee performance due to its low data rate.

A. Performance under different WiFi settings

To evaluate the performance under different WiFi devices settings with the default topology, we adjust the data rates and packet length of WiFi transmissions that may affect the ZigBee PRR. We let the WiFi AP sending data in 54Mbps as default, then in other standard rates like 48Mbps, 36Mbps, 24Mbps, 18Mbps and 12Mbps. Fig. 13(a) depicts how the WiFi throughput and ZigBee PRR are affected by the WiFi data rate. When the WiFi data rate increases, the transmission duration of a packet would be shorter, and the ZigBee device would get more chances to transmit packets. The black dashed line shows an extreme case that, under 24Mbps, the ZigBee PRR would be lower than 3%. However, with E-CCA, the PRR boosts to 30%, with about 35% WiFi throughput loss. When the data rate increases to 48Mbps, the WiFi packet is more vulnerable to collision. So, under high data rates, E-CCA exhibits comparable WiFi performance with the standard. Compared the solid PRR line with the dashed line, we could draw a conclusion that the PRR would improve 2x on average.

Fig. 13(b) shows the performance variation in terms of the WiFi packet length. We decrease the packet length by 100 bytes each time, and the data rate is fixed to 54Mbps. We see that increasing the packet length would improve the WiFi throughput but slightly degrade the ZigBee PRR due to the longer WiFi transmission duration. Meanwhile, E-CCA can largely increase the ZigBee PRR with little effect on the WiFi throughput. For example, with the packet length of 1500 bytes, E-CCA increases the WiFi PRR from about 22% to nearly 45%, while the WiFi throughput has only 0.1Mbps degradation.

B. Performance under different topology settings

We then investigate the influence of different topology settings. As shown in Fig. 12, the transmitter-receiver distance of WiFi and ZigBee links are denoted by d_w and d_z respectively, the distance between WiFi AP and ZigBee Tx is denoted by d_{wz}.

Now we adjust d_w by 1m every time and keep other distances fixed. It is obvious that d_w would not affect ZigBee PRR too much. From Fig. 14(a), we see that the PRR of standard ZigBee fluctuates at about 20% when d_w varies, but PRR of E-CCA is fixed at about 43% since d_{wz} is not changed. When d_w is shorter than 3m, E-CCA makes WiFi sacrifice its throughput for ZigBee transmission.

Next we set d_w as 5m and change d_z. As shown in Fig. 14(b), Under 2m, the SNR at ZigBee Rx is sufficient to decode data successfully and PRR could be 100%. With E-CCA, the PRR could increase from 90% to nearly 99% in 3m and boost 2x if the distance is more than 4m. The throughput loss of WiFi keeps static from 9.29126Mbps to 9.1476Mbps no matter what the distance d_z is.

Fig. 14(c) shows how d_{wz} affect the performance, while d_w and d_z are both fixed at 5m. When d_{wz} is shorter than 2m, with E-CCA, the PRR increases from 40% to nearly 100% through the channel reservation by WiFi. If the d_{wz} is more than 3m, The WiFi transmissions dominate the channel and make ZigBee PRR lower than 20%. In this situation, E-CCA doubles the PRR to more than 40% with nearly no sacrifice to WiFi throughput.

VII. Conclusion

In this paper, we propose the enhanced CCA (E-CCA) mechanism to mitigate the cross-technology interference, so as to improve the performance and fairness among heterogeneous wireless networks. Combined the traditional backoff mechanism with deep learning, the interference induced by power imbalance can be largely reduced and ZigBee devices can compete for the channel more fairly. E-CCA can be deployed easily since it only has simple modification on WiFi AP. We have set up testbed to collect data samples and test the identification accuracy accordingly. We have also conducted simulations based on NS-3 to show that E-CCA can increase ZigBee performance dramatically with the little throughput degradation for WiFi.

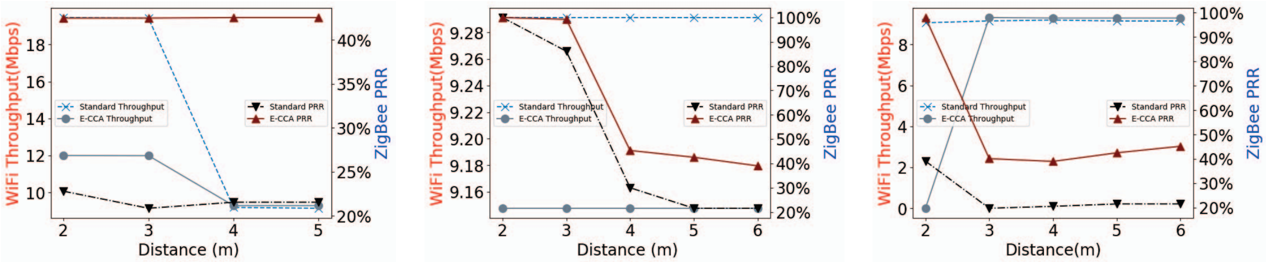
Acknowledgments

This research was supported in part by the China NSFC Grant (62072317), Guangdong NSF 2017A030312008, Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (161064), Shenzhen Science and Technology Foundation (ZDSYS20190902092853047), Faculty Research Fund of Shenzhen University (2019052, 860/000002110322), GDUPS (2015). Junmei Yao is the corresponding author.

References

[1] Inc Gartner , "Forecast: Internet of things — endpoints and associated services, worldwide." Website, 2017. https://www.gartner.com/en/documents/3840665.

[2] IEEE Computer Society. 802.11, "Wireless LAN medium access control (MAC) and physical layer



(a) the effect of d_w (b) the effect of d_z (c) the effect of d_{wz}

Fig 14. Performance under different topology settings

(PHY) specifications amendment 5: Enhancements for higher throughput," 2009.

[3] IEEE Computer Society. 802.15.4, "IEEE Standard for Low-Rate Wireless Networks," 2016.

[4] M. Samaneh, D. B. Smith, M. Abolhasan, and A. Jamalipour, "Opportunistic spectrum allocation for interference mitigation amongst coexisting wireless body area networks," ACM Transactions on Sensor Networks, 2018.

[5] Y. Yan, P. Yang, X. Li, and Y. Zhang, "Coffee: coexist wifi for zigbee networks: a frequency overlay approach," in Proc. of the ACM Tur-C, 2017.

[6] Z. Yin, Z. Li, S. M. Kim, and T. He, "Explicit channel coordination via cross-technology communication," in Proc. of the ACM MobiSys, 2018.

[7] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J.Ma, "Zifi: Wireless lan discovery via zigbee interference signatures," in Proc. of the ACM MobiCom, 2010.

[8] S. Rayanchu, A. Patro, and S. Banerjee, "Airshark:detecting non-wifi rf devices using commodity wifi hardware," in Proc. of the ACM IMC, 2011.

[9] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "Zisense: Towards interference resilient duty cycling in wireless sensor networks," in Proc. of the ACM SenSys, 2014.

[10] J. Meng, Y. He, X. Zheng, D. Fang, D. Xu, T. Xing, and X. Chen, "Smoggy-link: Fingerprinting interference for predictable wireless concurrency," in Proc. of the IEEE ICNP, 2016.

[11] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L. Nordén, and P. Gunningberg, "Sonic: classifying interference in 802.15.4 sensor networks," in Proc. of the IEEE IPSN, 2013.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 1998.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," 2012.

[14] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over the air deep learning based radio signal classification," IEEE Journal of Selected Topics in Signal Processing, 2017.

[15] A. Sikora and V. F. Groza, "Coexistence of ieee802.15.4 with other systems in the 2.4 ghz-ism-band," in Proc. of the IEEE I2MTC, 2006.

[16] R. Musaloiu-E. and A. Terzis, "Minimising the effect of wifi interference in 802.15.4 wireless sensor networks," International Journal of Sensor Networks, 2007.

[17] X. Zhang and G. S. Kang, "Enabling coexistence of heterogeneous wireless systems: Case for zigbee and wifi," in Proc. of the ACM MobiHoc, 2011.

[18] R. Božidar, R. Chandra, and D. Gunawardena, "Weeble: Enabling low-power nodes to coexist with high-power nodes in white space networks," in Proc. of the ACM CoNEXT, 2012.

[19] C. J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving wi-fi interference in low power zigbee networks," in Proc. of the ACM SenSys, 2010.

[20] Y. Wang, Q. Wang, Z. Zeng, G. Zheng, and R. Zheng, "Wicop: Engineering wifi temporal whitespaces for safe operations of wireless body area networks in medical applications," in Proc. of the IEEE RTSS, 2011.

[21] Z. Li and T. He, "Webee: Physical-layer cross-technology communication via emulation," in Proc. of the ACM MobiCom, 2017.

[22] W. Wang, T. Xie, X. Liu, and T. Zhu, "Ect: Exploiting cross-technology transmission for reducing packet delivery delay in iot networks," ACM Transactions on Sensor Networks, 2019.

[23] R. Chen and W. Gao, "Enabling cross-technology coexistence for extremely weak wireless devices," in Proc. of the IEEE INFOCOM, 2019.

[24] T. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in Proc. of the GRCon, 2016.

[25] N. P. Jouppi, C. S. Young, et al., "In-datacenter performance analysis of a tensor processing unit," in Proc. of the ACM ISCA, 2017.