AP-Assisted CTS-Blocking for WiFi-ZigBee Coexistence

Shigemi Ishida*, Shigeaki Tagashira†, Akira Fukuda*
*Faculty of Information Science and Electrical Engineering, Kyushu University, Japan
Email: {ishida, fukuda}@f.ait.kyushu-u.ac.jp
†Faculty of Informatics, Kansai University, Japan
Email: shige@res.kutc.kansai-u.ac.jp

***Abstract*—WiFi interference is one of the big problems in ZigBee-based sensor networks. In this paper, we present a new WiFi-ZigBee coexistence scheme named *AP-assisted CTS-blocking* (AA CTS-blocking). The AA CTS-blocking uses an RTS/CTS (request to send, clear to send) mechanism to prevent WiFi transmissions during ZigBee communications. An RTS/CTS mechanism is defined in the IEEE 802.11 standard and is supported by off-the-shelf WiFi devices. We present the design and implementation of AA CTS-blocking utilizing an off-the-shelf WiFi device as it is. The experimental evaluations revealed that AA CTS-blocking reduced frame error rate (FER) by approximately 5 % compared to an existing WiFi-ZigBee coexistence scheme.**

***Index Terms*—WiFi, ZigBee, collision avoidance, hidden terminal problem, CTS-blocking.**

I. Introduction

Sensor network is gaining importance due to its low-cost and low-power features in the fields of machine-to-machine (M2M) communications, Internet of Things (IoT), and Cyber Physical Systems (CPS). Sensor nodes are usually equipped with low-power IEEE 802.15.4 (ZigBee) modules that work in a 2.4-GHz ISM (Industry, Scientific, and Medical) band.

A 2.4-GHz ISM band is used by many wireless technologies such as WiFi and Bluetooth because the ISM band is legally available as an unlicensed band. WiFi and Bluetooth are widely used today in many indoor environments, which interfere with ZigBee communications. Especially, WiFi communications highly affect ZigBee communications due to higher transmission power and shorter frame length compared to ZigBee. ZigBee channels overlap with WiFi channels as shown in Fig. 1, which makes the coexistence problem more severe [1].

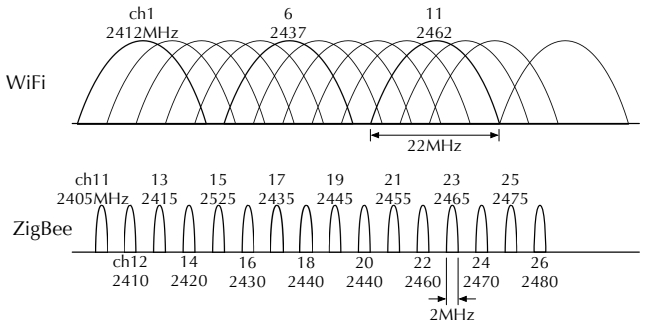

Fig. 1. WiFi and ZigBee channels

To reduce WiFi interference in ZigBee communications, studies on WiFi-ZigBee coexistence have been conducted. These studies primarily rely on an existing CSMA (carrier sense multiple access) mechanism [2]–[5] or WiFi traffic statistics [6]. Although these studies have successfully reduced the WiFi interference in ZigBee communications, the studies put some impractical requirements such as special hardwares, controls of all WiFi APs, no significant change of WiFi traffic, and OS modifications. These requirements make it impractical to implement the coexistence schemes.

In view of this, we present AP-assisted CTS-blocking (AA CTS-blocking) employing an off-the-shelf WiFi device to control WiFi communications. We send an RTS (request to send) frame to a WiFi AP prior to ZigBee communication. An RTS frame lets the AP to transmit a CTS (clear to send) frame, which suppresses WiFi transmissions for specific duration. The AA CTS-blocking is an extension of CTS-blocking [2] with no OS modification.

By implementing and evaluating a data collection system utilizing AA CTS-blocking using actual sensor nodes, we show the effectiveness of AA CTS-blocking despite its simple design. Specifically, our main contributions are threefold:

- We propose a new WiFi-ZigBee coexistence scheme named AP-assisted CTS-blocking (AA CTS-blocking). Utilizing a WiFi AP in the environment, we can build a cross-technology collision avoidance scheme with off-the-shelf WiFi device with no modification.
- We present a design and implementation of a sensor network system employing AA CTS-blocking. Our simple design is easily adapted to a simple time division multiple access (TDMA) data collection system.
- We conducted experimental evaluations and showed that the AA CTS-blocking effectively reduced frame error rate compared to an existing WiFi-ZigBee coexistence scheme.

The remainder of this paper is organized as follows. In Section II, we briefly look through related works. Section III describes the design of AA CTS-blocking. We implemented a data collection system using AA CTS-blocking in Section IV and evaluated the system in Section V to show the effectiveness of AA CTS-blocking. Section VI concludes the paper.

II. Related Works

Several studies have reported on a WiFi-ZigBee coexistence problem.

doi: 10.1109/CANDAR.2015.16

Han et al. [7] analyzes ZigBee error rate in WiFi-ZigBee coexistence scenarios. Combining appropriate channel planning with an appropriate clear channel assessment (CCA) mode, ZigBee error rate can be minimized to less than 5 %. Appropriate channel planning, however, is often unrealistic because too many WiFi APs are already installed in many environments.

CBT [4] utilizes special ZigBee nodes named *signaler*s to kick a CSMA mechanism on WiFi devices. The signalers are installed beside all WiFi devices and continuously transmit dummy ZigBee frames during ZigBee communications. The signaler transmits dummy frames in a channel different from ZigBee communication channel. WiFi devices detect ZigBee signals from signalers and refrain from transmissions based on a CSMA mechanism. CBT is powerful but impractical for mobile WiFi devices.

CTS-blocking [2] uses CTS (clear to send) frames to suppress WiFi transmissions during ZigBee communications. WiFi devices that receive a CTS frame refrain from sending for specific duration. During this WiFi-blocked period, ZigBee nodes freely communicate each other. Dynamic GTS [3] also uses the CTS-blocking scheme. In dynamic GTS, IEEE 802.15.4 GTS (guaranteed time slots) are protected using CTS-blocking scheme. Unfortunately, recent WiFi devices except APs are not allowed to send CTS frames in terms of communication fairness. We need to modify OS to send CTS frames.

CACCA [5] empowers a clear channel assessment (CCA) function by employing a special hardware or modifying OS to detect signals of other wireless technologies. However, CACCA requires all wireless devices to employ the empowered CCA function, which is often unrealistic.

WISE [6] analyzes WiFi traffic pattern and find *white space*s for ZigBee communications. ZigBee nodes divide data into short frames that fit to white spaces and transmit the sequence of short frames. WISE is based on WiFi traffic statistics and is therefore delicate with WiFi traffic change.

BuzzBuzz [8] employs error correction techniques to recover corrupted ZigBee frames. BuzzBuzz send a header multiple times and append Reed Solomon error correction code to payloads. Error correction approach can be combined with our AA CTS-blocking.

III. AP-Assisted CTS-Blocking

AP-assisted CTS-blocking is an extension of CTS-blocking reported in [2]. In the following subsections, we first describe an overview of CTS-blocking and then extend the CTS-blocking to the AP-assisted CTS-blocking.

A. CTS-blocking

Figure 2 depicts an overview of CTS-blocking. The CTS-blocking system consists of a controller, a ZigBee coordinator wired to the controller, and ZigBee end devices. CTS-blocking utilizes a controller employing a WiFi module to send CTS frames to block WiFi transmissions.

To start ZigBee communication, the controller transmits a CTS frame. WiFi devices that receive the CTS frame stop transmissions for the duration specified by a `Duration` field in the CTS frame. We call this period with no WiFi transmissions as *WiFi-blocked period*. During the WiFi-blocked period, ZigBee nodes freely communicate each other.

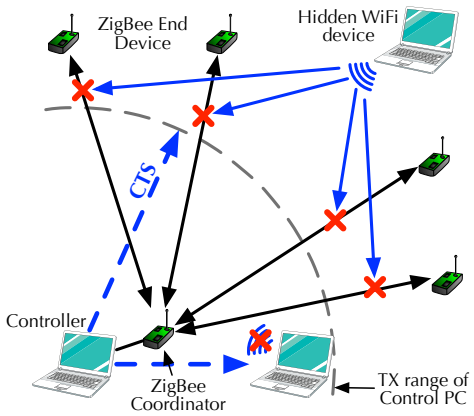

Fig. 2. Overview of CTS-blocking as well as hidden terminal problem

CTS-blocking is a simple scheme and faces a CTS transmission problem; WiFi devices nowadays except APs are limited to transmit CTS frames in terms of communication fairness. Moreover, due to a dynamic power control of WiFi devices, CTS-blocking has higher chance to be suffered from hidden terminals as shown in Fig. 2.

B. AP-assisted CTS-blocking

Figure 3a shows an overview of the proposed AP-assisted CTS-blocking (AA CTS-blocking). AA CTS-blocking utilizes a WiFi AP, named a helper AP, installed in the environment to address the CTS transmission problem; we send RTS frames from a controller to let the helper AP to send CTS frames.

Figure 3b depicts the communication sequence of AA CTS-blocking. To initiate ZigBee communication, 1) the controller chooses an AP. We call this AP as a helper AP. 2) The controller transmits an RTS frame to the helper AP. 3) Receiving the RTS frame, the helper AP broadcasts a CTS frame to WiFi devices. WiFi devices that receive the CTS frame refrain from transmission, which results in blocking of WiFi communication for specific duration. 4) After receiving the CTS frame, the controller sends a signal to ZigBee coordinator to start ZigBee communication. ZigBee coordinator controls ZigBee communication during the WiFi-blocked period.

Implementation of the proposed AA CTS-blocking brings two practical design issues below:

1) How to choose a helper AP?: There are many WiFi APs installed in the environment today. We need to choose a helper AP, i.e., a destination AP of an RTS frame. Helper AP selection has an effect on performance of the proposed AA CTS-blocking because CTS frames are sent from the helper AP with limited transmission power resulting in limited coverage of WiFi blocking.

2) How to schedule ZigBee communication to complete within a WiFi-blocked period?: Using a `Duration` field in CTS frames, AA CTS-blocking blocks WiFi transmissions. The duration of WiFi-blocked periods is limited to maximum of approximately 32 milliseconds. We need to optimize ZigBee communication schedule to maximize throughput.

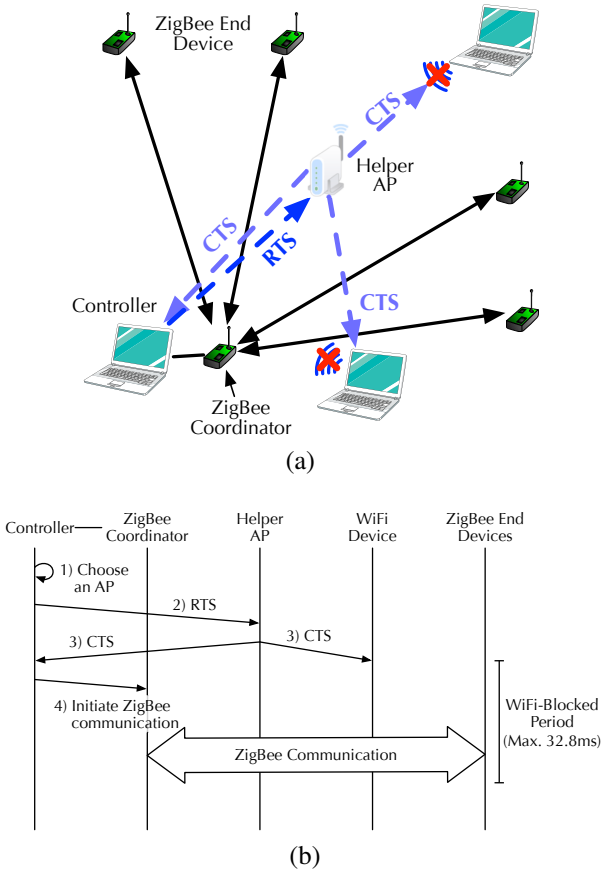

Fig. 3. AP-assisted CTS-blocking: (a) overview, (b) communication sequence

Assuming that a ZigBee network is a single-hop data collection network, we present a simple AP-selection algorithm and example scheduling algorithm in Sections III-C and III-D, respectively.

C. Helper AP Selection

To choose a helper AP, we need to maximize overlap of coverage between a helper AP and ZigBee network. Maximum coverage overlapping implies that the number of WiFi devices in ZigBee coverage is maximized, which effectively reduces WiFi interference in a ZigBee network.

Without location information of APs and ZigBee end devices, maximizing the coverage overlapping is impractical. Assuming that the ZigBee network is a single-hop data collection network, we choose the nearest AP from the controller as a helper AP to achieve quasi-maximum overlapping. The controller and ZigBee coordinator are not distant because the controller is wired to the ZigBee coordinator. The coverage of the nearest AP and ZigBee network therefore have close centers with different radii.

To estimate distance between each AP and the controller, we utilize received signal strength (RSS). The controller periodically scans on WiFi channels and collects AP beacon frames retrieving operating channel information and an RSS. The controller chooses the AP whose beacon RSS is the biggest.

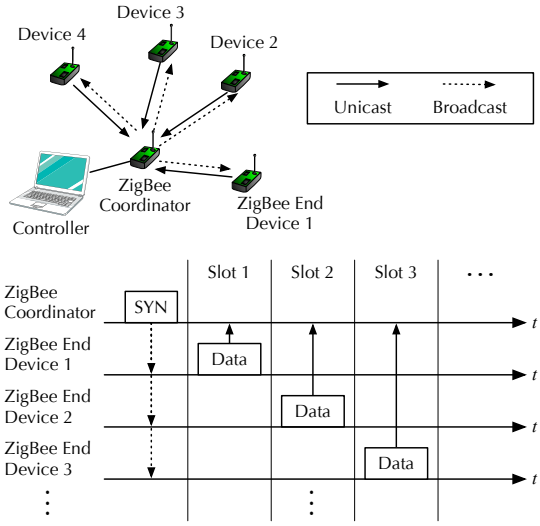

Fig. 4. Example TDMA-based scheduling of ZigBee communication. Each ZigBee end device sends sensor data in a specific time slot.

In the proposed AP selection, we assume that the ZigBee channel is predefined. Although there are studies on dynamic channel adaptation for a ZigBee-WiFi interference problem [9], [10], this assumption is natural because majority of channel-adaptation schemes decide the channel prior to their communication.

D. ZigBee Communication Scheduling

The maximum value of `Duration` fields is defined as 32,767 (in microseconds) in the IEEE 802.11 standard. The duration of WiFi-blocked periods is therefore limited to a maximum of 32.767 milliseconds.

As long as ZigBee communications complete within 32.767 milliseconds, we can use any scheduling algorithm for ZigBee communications. To show an example, we assume that the ZigBee network is used as a sensor data collection network; each ZigBee node retrieves sensor data and sends the data to a sink node, i.e., the ZigBee coordinator.

For a data collection network, we schedule ZigBee communications using a simple TDMA-based MAC protocol as shown in Fig. 4. A ZigBee coordinator broadcasts a synchronization frame immediately after receiving a signal from a controller. Each ZigBee end device sends sensor data to the ZigBee coordinator in a slot specified by the device ID.

Practically, the number of slots is limited to few tens due to the limited duration of WiFi-blocked period. We group ZigBee nodes and one group is allowed to transmit data in each WiFi-blocked period.

IV. Implementation

To conduct experimental evaluations, we implemented a data collection sensor network utilizing AA CTS-blocking depicted in Fig. 4. We used a controller laptop and 11 MICAz ZigBee nodes from Crossbow.

The controller program was implemented as a C program on Debian GNU/Linux 8.0 running on the laptop. The controller

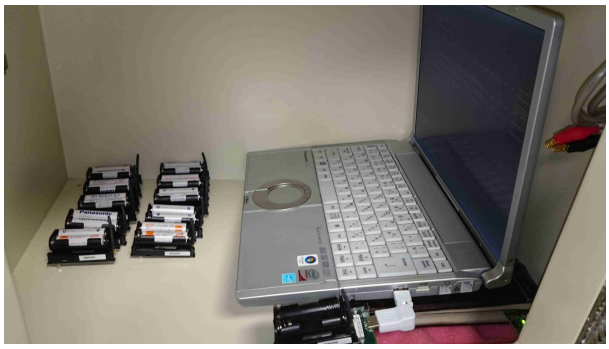

Fig. 5. Preliminary experiment setup. A controller laptop, ZigBee nodes, and a CC2531 USB dongle were installed in an anechoic box. ZigBee coordinator was the node wired to the controller.

used a `libpcap` library to transmit RTS frames and to receive CTS frames.

For helper AP selection, the controller also used a `libpcap` library. Prior to ZigBee communication, the controller sniffed all the WiFi beacon frames on a monitor mode interface using a `libpcap` library for 1.5 seconds. The controller then analyzed the WiFi frames with a Radiotap header to retrieve AP information and selected a helper AP as described in Section III-C.

A ZigBee network was implemented as a user-space application on TinyOS rather than a modified MAC protocol. A MICAz was used as a ZigBee coordinator, which was wired to the controller laptop. The other ten MICAzs were used as ZigBee end devices. Each ZigBee end device generated dummy data and sent the data to the ZigBee coordinator in a specific slot during a WiFi-blocked period. A dummy data size was 18 bytes including an IEEE 802.15.4 header. A dummy data transmission took 0.576 milliseconds because the IEEE 802.15.4 transmission rate in 2.4 GHz is 250 kbps.

V. Experimental Evaluation

To demonstrate the effectiveness of the proposed AA CTS-blocking, we conducted experimental evaluations. We first performed a preliminary experiment and determined a slot size for the TDMA-based scheduling described in Section III-D. We then evaluated the frame error rate (FER) of a data collection system utilizing AA CTS-blocking under WiFi environment.

A. Preliminary Experiment

In Section IV, a data collection system utilizing AA CTS-blocking was implemented as a user-space application on TinyOS. A slot size was unable to be controlled precisely because of interruptions and OS processes. To determine a slot size, we evaluated frame error rate (FER) in an RF anechoic box that is insulated exterior radio signals.

Figure 5 shows a preliminary experiment setup. We installed a controller laptop (CF-Y8 from Panasonic) wired to a MICAz ZigBee coordinator and ten MICAz ZigBee end devices in an RF anechoic box. A CC2531 USB dongle from Texas Instruments was also installed to monitor all the IEEE 802.15.4 transmissions.

Each ZigBee end device transmitted dummy data to the ZigBee coordinator in a specific slot. Data collection from all the ten ZigBee end devices was repeated for 200 times. We varied a slot size from 1 to 3 milliseconds and evaluated FER.

TABLE I
Data collection FER in an RF anechoic box

Slot size	Frame Error Rate (FER)
1 ms	50.2 %
2 ms	0.05 %
3 ms	0.07 %

Table I shows the FER of data collection in an RF anechoic box. We confirmed that increase in a slot size resulted in decrease in FER. When slot size was greater or equal to 2 milliseconds, FER was saturated. We therefore used a slot size of 2 milliseconds.

B. Experiment Setup

Figure 6 depicts an experiment setup. We installed a controller laptop wired to a ZigBee coordinator, ten ZigBee end devices, and six WiFi APs in our laboratory. We also installed five WiFi devices that generate 5 Mbps traffic in total, which was monitored by a Wireshark network protocol analyzer. ZigBee and WiFi channels were configured to 18 and 6, respectively. These channels overlap as shown in Fig. 1.

We define a collection sequence as the data collection from all the ten ZigBee end devices using AA CTS-blocking. Each sequence, ten ZigBee end devices transmitted dummy data of 18 bytes to a ZigBee coordinator in a specific slot. Collection sequences were initiated from a controller every 200 milliseconds and repeated for 1,000 times.

In order to show the relative performance, we compared three schemes below:

1) *No control:* ZigBee nodes freely communicated.
2) *CTS-blocking:* A controller transmitted a CTS frame prior to ZigBee communication.
3) *AA CTS-blocking:* The proposed method. A controller transmitted an RTS frame prior to ZigBee communication to let a helper AP to transmit a CTS frame.

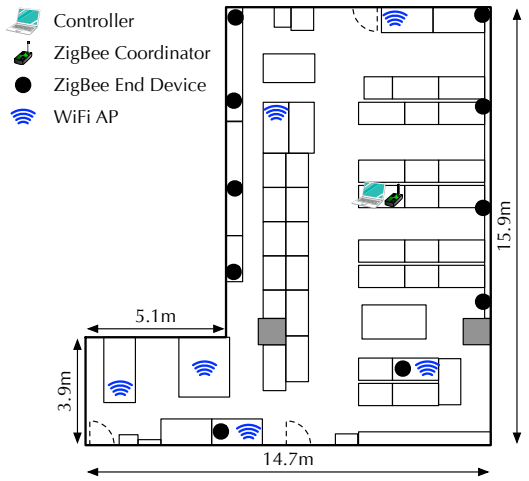

Fig. 6. Experiment setup

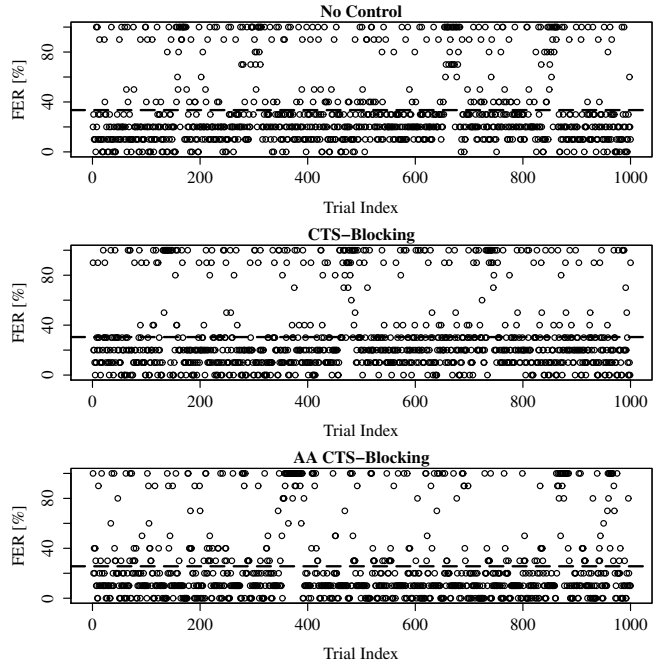

Fig. 7. Frame error rate (FER) of each trial; dashed lines are average FER over all trials

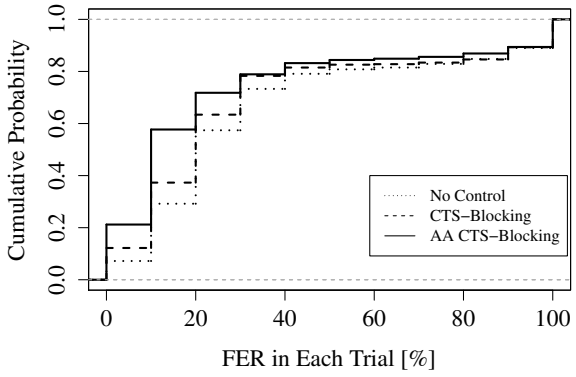

Fig. 8. Empirical cumulative distribution function of frame error rate (FER) in each trial

Note that we modified a MAC protocol on Debian GNU/Linux for CTS-blocking to directly transmit CTS frames from a controller.

C. Frame Error Rate

Figure 7 shows the ZigBee frame error rate (FER) of each trial; dashed lines are average FER. The average FER of the no control, CTS-blocking, and AA CTS-blocking schemes were 33.48 %, 30.46 %, and 25.60 %, respectively. The AA CTS-blocking exhibited the smallest average FER among the three schemes. The FER was reduced by 7.88 % and 4.86 % compared to the no control and CTS-blocking schemes, respectively. AA CTS-blocking successfully suppressed WiFi transmissions during ZigBee communication, which reduced frame errors. In an AA CTS-blocking scheme, CTS frames were sent from a helper AP without dynamic power control resulting in reduced influence of hidden terminals.

Although the AA CTS-blocking scheme showed low average FER, there were many frame errors as shown in Fig. 7. To confirm the effectiveness of AA CTS-blocking of each trial, we examined the frame error rate (FER) of each trial.

Figure 8 shows empirical cumulative distribution function of the FER of each trial. In an AA CTS-blocking scheme, the FER was less than 20 % for more than 50 % of trials. Low FER of each trial resulted in low average FER. In all the three schemes, FER was 100 % for approximately 10 % of trials. When a WiFi network was too congested, all the three schemes failed to overcome WiFi interference.

VI. Conclusion

In this paper, we presented AP-assisted CTS-blocking (AA CTS-blocking) for coexistence of WiFi and ZigBee networks. AA CTS-blocking uses an off-the-shelf WiFi device as well as a WiFi AP installed in the environment to suppress WiFi transmissions, which reduces ZigBee frame errors. We implemented a data collection system employing AA CTS-blocking and conducted experimental evaluations. The experimental results demonstrated that the AA CTS-blocking successfully reduced frame error rate by approximately 5 % compared to an existing WiFi-ZigBee coexistence scheme.

Acknowledgment

This work was supported in part by JSPS KAKENHI Grant Number 25870928.

References

[1] K. Shuaib, M. Boulmalf, F. Sallabi *et al.*, "Co-existence of Zigbee and WLAN, a performance study," in *Proc. IEEE Wireless Telecommunications Symposium (WTS)*, Apr. 2006, pp. 1–6.

[2] J. Hou, B. Chang, D.-K. Cho *et al.*, "Minimizing 802.11 interference on Zigbee medical sensors," in *Proc. Int. Conf. Body Area Networks (BodyNets)*, Apr. 2009, pp. 1–8.

[3] M. L. Huang and S.-C. Park, "A WLAN and ZigBee coexistence mechanism for wearable health monitoring system," in *Proc. Int. Symp. Communications and Information Technology (ISCIT)*, Sep. 2009, pp. 555–559.

[4] X. Zhang and K. G. Shin, "Enabling coexistence of heterogeneous wireless systems: Case for ZigBee and WiFi," in *Proc. ACM MobiHoc*, May 2011, pp. 1–11.

[5] L. Tytgat, O. Yaron, S. Pollin *et al.*, "Avoiding collisions between IEEE 802.11 and IEEE 802.15.4 through coexistence aware clear channel assessment," *EURASIP J. Wireless Communications and Networking*, vol. 2012, no. 137, pp. 1–15, Apr. 2012.

[6] J. Huang, G. Xing, G. Zhou *et al.*, "Beyond co-existence: Exploiting WiFi white space for ZigBee performance assurance," in *Proc. IEEE Int. Conf. Network Protocols (ICNP)*, Oct. 2010, pp. 305–314.

[7] T. Han, B. Han, L. Zhang *et al.*, "Coexistence study for WiFi and ZigBee under smart home scenarios," in *Proc. IEEE Int. Conf. Network Infrastructure and Digital Content (IC-NIDC)*, Sep. 2012, pp. 669–674.

[8] C.-J. Liang, N. B. Priyantha, J. Liu *et al.*, "Surviving Wi-Fi interference in low power ZigBee networks," in *Proc. ACM SenSys*, Nov. 2010, pp. 309–322.

[9] E. T. Yazdi, A. Willig, and K. Pawlikowski, "Coupling power and frequency adaptation for interference mitigation in IEEE 802.15.4-based mobile body sensor networks," in *Proc. IEEE Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, Apr. 2014, pp. 1–6.

[10] L. Tang, Y. Sun, O. Gurewitz *et al.*, "EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks," in *Proc. ACM MobiHoc*, May 2011, pp. 1–11.