Design of WiFi AP-RSS Monitoring System using Sensor Nodes

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

***Abstract*—Sensor localization is one of the big problems when building large scale indoor sensor networks because GPS is unavailable in indoor environments. In this paper, we propose a sensor localization system using WiFi APs as anchors. WiFi APs are largely installed in indoor environments and are managed by a network system manager. Using WiFi APs as anchors, we can localize sensor nodes with neither newly deployed anchor nodes nor user cooperation.**

As a first step of our sensor localization system, this paper presents a WiFi AP-RSS monitoring system using sensor nodes. Sensor nodes are equipped with ZigBee (IEEE 802.15.4) modules, which cannot demodulate WiFi (IEEE 802.11) signals. We therefore developed a *cross-technology signal extraction scheme* with an AP recognition scheme on sensor nodes. We herein describe the design and implementation of our AP-RSS monitoring system. The experimental evaluations demonstrated that the proposed AP-RSS monitoring system successfully recognized WiFi APs with detection errors less than 10 %. We also confirmed that the proposed system monitored AP-RSS with an average error of 1.26 dB.

***Index Terms*—WiFi AP anchors, sensor localization, cross-technology communication.**

I. Introduction

Sensor networks play an important role in many fields such as security, agriculture, fishing, forestry, construction, transportation, and environmental monitoring. In sensor networks, sensor location is important for recognizing sensing area, target tracking, and a network routing. Building sensor network systems always require to localize all the sensor nodes.

Sensor location is usually derived by using global positioning system (GPS) or manual measurements. We face a sensor localization problem when we build a large scale sensor network in an indoor environment, where GPS is unavailable. The BEMS (Building Energy Management System) and a security system are typical examples of large scale indoor sensor network systems. The sensor localization problem prevents sensor networks from becoming more prevalent.

To mitigate sensor localization problem, there have been so much literature reporting indoor localization systems [1–3]. These studies have primarily investigated reduction in deployment costs [4–13] or accuracy improvement [14–20]. Although these studies have successfully reduced the cost of sensor localization, they require user cooperation or anchor nodes whose location is manually measured.

Our goal is to realize an indoor sensor localization system that requires no newly deployed anchor nodes. In this paper, we propose an indoor sensor localization system using WiFi APs as anchors. WiFi APs are largely installed in many indoor environments and their locations are managed by a network system manager. We send specific signals from multiple WiFi APs and monitor the received signal strength (RSS) on sensor nodes. We then calculate locations of sensor nodes using an RSS-based localization scheme such as triangulation.

As a first step of the sensor localization system using WiFi APs, this paper presents a WiFi AP-RSS monitoring system using sensor nodes. Sensor nodes are equipped with ZigBee (IEEE 802.15.4) modules and cannot demodulate WiFi signals. To measure RSS of WiFi signals on sensor nodes, we developed a cross-technology signal extraction scheme. In this scheme, we employ a signal folding technique presented in ZiFi [21] and retrieve AP-RSS with a simple filtering method.

In our previous work, we presented an AP-RSS monitoring scheme with a simple AP recognition using non-overlapping channels [22]. Although the AP recognition effectively works in some environments, recent dense WiFi deployment makes it difficult to use non-overlapping channels on each AP. In this paper, we extend our previous work to safely distinguish APs; beacon intervals with specific constraints are used as AP IDs.

By implementing the RSS monitoring system using a sensor node MICAz, we show the feasibility of our sensor localization system. We also conduct experiments to show that our monitoring system has enough performance to build a localization system.

Specifically, our main contributions are fourfold:

- We propose a new indoor sensor localization system that uses WiFi APs as anchors. Using WiFi APs, we can localize sensor nodes without newly deployed anchors.
- We present the design of a WiFi AP-RSS monitoring system for sensor nodes employing a ZigBee (IEEE 802.1.5) module. Our design is based on an existing signal processing technique combined with a signal recognition method. We also apply a simple filtering method to mitigate some practical issues for RSS measurement.
- We propose an accurate AP recognition scheme, which is supported by a mathematical theorem. We conducted experiments to show the accuracy of our AP recognition scheme.

doi: 10.1109/CANDAR.2015.54

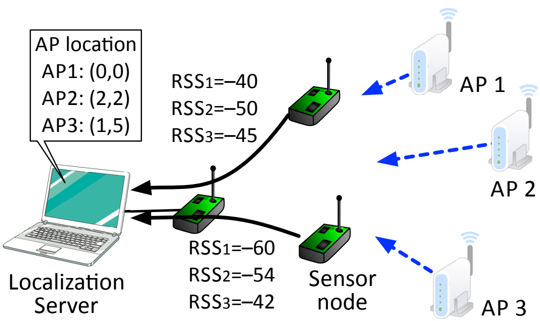

Fig. 1. Overview of a sensor localization system using WiFi APs as anchors

- We show the feasibility of our sensor localization system by experimental evaluations of the AP-RSS monitoring system using an actual sensor node MICAz.

The remainder of this paper is organized as follows. Section II describes our new sensor localization system as well as design challenges. Section III designs an AP-RSS monitoring system. In Section IV, we implemented our AP-RSS monitoring system using an actual sensor node. Experiments were conducted in Section V to evaluate the basic performance. We briefly look through related works in Section VI. Section VII concludes the paper.

II. Sensor Localization System using WiFi APs

Figure 1 depicts an overview of a sensor localization system using WiFi APs as anchor nodes. The sensor localization system consists of sensor nodes, a localization server, and WiFi APs installed in the environment. WiFi APs periodically transmit a specific signal that can be detected by sensor nodes.

To initiate localization process, sensor nodes first detect AP signals and measure received signal strength (RSS). The sensor nodes then send the RSS information to a localization server. Using the RSS information, the localization server calculates sensor location by a localization scheme such as multilateration. Here we assume that the localization server holds AP location data. This assumption is natural because the APs are usually managed by a network system manager.

Two challenges come up toward realizing the sensor localization system using WiFi APs.

1) How to detect WiFi AP signals on sensor nodes?: Sensor nodes cannot demodulate WiFi (IEEE 802.11) signals since sensor nodes are equipped with ZigBee (IEEE 802.15.4) modules. We need to pick WiFi AP signals out from many WiFi device signals. We then measure RSS of the AP signals.

2) How to recognize sender APs?: Using the RSS information, we can calculate distance between a sensor node and an AP. For sensor localization, we need to associate the distance with a specific sender AP.

In the following section, we design an AP-RSS monitoring system that tackles the above two challenges.

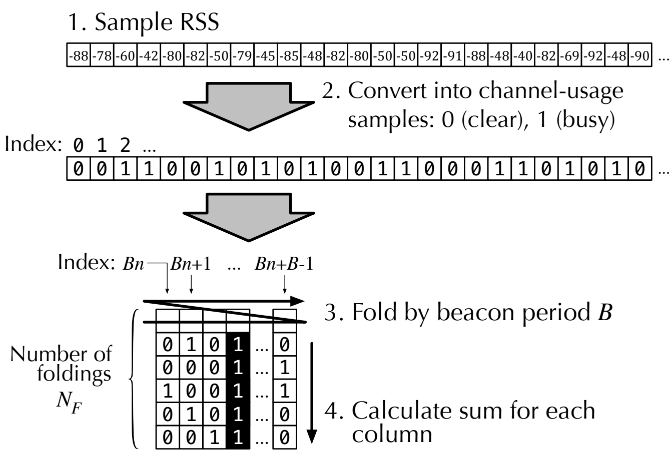

Fig. 2. AP signal detection by folding. 1) A sensor node periodically samples RSS, 2) and convert the RSS samples into channel-usage samples. 3) The sensor node folds the channel-usage samples by beacon period, 4) and calculate a sum for each column to get channel-usage sums. Periodic beacon signals appear in a same column, which results in a large channel-usage sum.

III. AP-RSS Monitoring System

A. Design Overview

The design of our AP-RSS monitoring system is divided into three sub-designs: AP signal detection, AP recognition, and AP-RSS extraction. To detect AP signals on sensor nodes, we employ a simple signal processing technique presented in ZiFi [21]. ZiFi is based on periodicity of AP beacon signals. We therefore configure each AP to have different beacon intervals with specific constraints to safely recognize sender APs. We finally retrieve AP-RSS with a simple filtering method based on beacon length.

The following three subsections give details of each sub-design.

B. AP Signal Detection

Figure 2 depicts a process of AP signal detection. To detect AP signals on a sensor node, the sensor node periodically samples RSS in a specific channel. Note that all ZigBee (IEEE 802.15.4) modules have an RSS measurement function as an energy detection function defined in the standard [23]. The sensor node can detect WiFi signals because both ZigBee and WiFi are using a 2.4-GHz band (Fig.3a). Figure 3b shows an example of WiFi signals on ZigBee channel 19 retrieved by a MICAz sensor node. Because ZigBee modules provide average RSS over 128 microseconds, which is defined in the standard, we set the sampling period to 128 microseconds not to miss WiFi signals while minimizing a sampling rate.

The sensor node converts the each RSS sample into a channel-usage sample: `0` for clear and `1` for busy. We use -77 dBm as a threshold for channel-usage determination. This threshold follows after the default threshold of a CC2420 IEEE 802.15.4 module for clear channel assessment [24].

We then fold the channel-usage samples by the AP beacon period and create a channel-usage matrix. Consider folding of beacon signals whose interval is t. Because time length of each

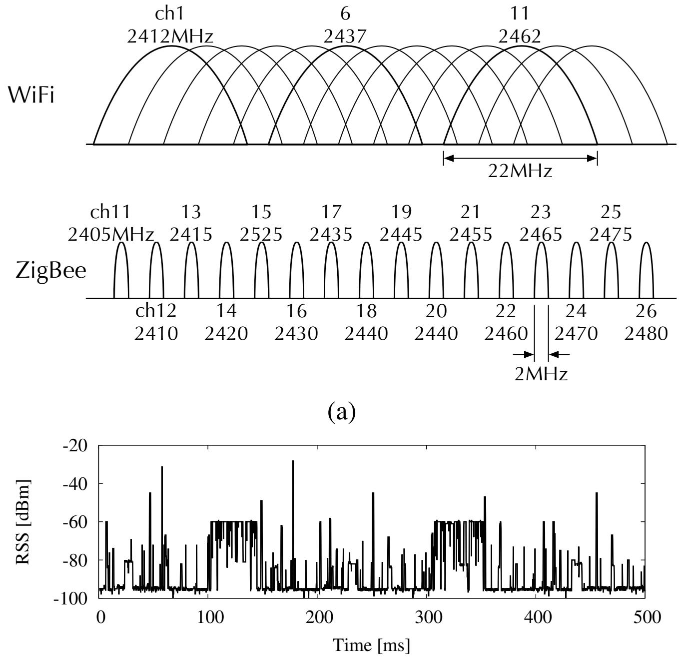

Fig. 3. (a) WiFi and ZigBee channels, and (b) WiFi signals on ZigBee channel 19 retrieved by MICAz

channel-usage sample is 128 microseconds, we can calculate a beacon period B as $t/(128 \times 10^{-6})$. We then calculate sum for each column in the channel-usage matrix. We name this sum as *channel-usage sum*.

We can detect APs by finding columns whose channel-usage sum is large enough. Beacon signals whose interval equals to the folding period appear in a same column. Large channel-usage sum therefore indicates that there are beacon signals whose interval equals to folding period, as shown in Fig. 2.

Because a channel-usage sum is related to the number of folding N_F, we cannot use a channel-usage sum itself; rather we use a *superposition-rate* defined as a ratio of channel-usage sum to N_F. Beacon signals are found when a superposition-rate is above a threshold. We conduct a preliminary experiment to determine the number of folding N_F and a superposition-rate threshold in Section V-A.

C. AP Recognition

To safely detect each AP on sensor nodes, we apply the signal multiplexing technique presented in FreeBee [25]. We configure APs to transmit beacon signals with different intervals that are *non-multiples* each other. We can easily recognize beacon signals having non-multiple intervals by the AP detection technique presented in the previous subsection. This simple recognition technique is supported by the following theorem.

Theorem 1. *Let t_A and t_B be positive integers. When we fold beacon signals of interval t_B by fold period t_A, the beacon signals appear in a same column in a channel-usage matrix at most every $\mathrm{lcm}(t_A, t_B)/t_A$ rows, where $\mathrm{lcm}(t_A, t_B)$ is the least common multiple of t_A and t_B.*

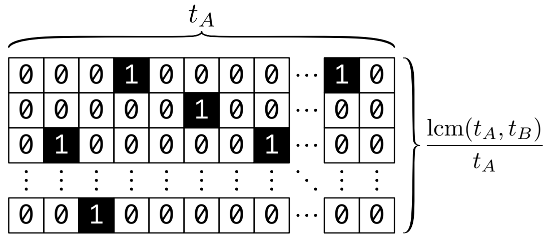

Fig. 4. Beacon signals of interval t_B and of length $\mathrm{lcm}(t_A, t_B)$ folded by period t_A

Proof: We prove by contradiction. Suppose beacon signals of interval t_B appear twice in a same column within $\mathrm{lcm}(t_A, t_B)/t_A$ rows. Length of channel-usage samples corresponding to $\mathrm{lcm}(t_A, t_B)/t_A$ rows is $\mathrm{lcm}(t_A, t_B)$, as shown in Fig. 4. Thus, positive integers m and n exist such that

$$m \cdot t_B = n \cdot t_A < \mathrm{lcm}(t_A, t_B). \tag{1}$$

This equation contradicts that $\mathrm{lcm}(t_A, t_B)$ is the least common multiple of t_A and t_B. Therefore, the theorem follows. ■

If t_A is not a multiple of t_B, $\mathrm{lcm}(t_A, t_B) > t_A$. Beacon signals of interval t_B therefore appear in a same column at most every two rows when we fold the beacon signals by fold period t_A. We can theoretically remove beacon signals of interval t_B using a superposition-rate threshold greater than 0.5.

We note that almost all the APs available today has default beacon interval of 100 TU (TU: time unit $= 1024$ microseconds, defined in the standard [26]). When we determine a beacon interval, we need to avoid multiples and divisors of 100.

In a real environment, beacon frames are delayed because of a CSMA/CA mechanism. Delayed beacons can be detected falsely as beacons of similar interval. To reduce false detection, we remove pairs of beacon intervals that can be falsely detected using algorithm 1. The function $p(A, B)$ in algorithm 1 is the probability that beacon signals of interval A are detected as interval B signals.

Configuring a beacon interval on all the WiFi APs is sometimes impractical. In this case, we have an option to use ZigBee channels to recognize sender APs; we sample RSS on different ZigBee channels and detect AP signals whose interval is 100 TU. Applying a signal detection scheme presented in the previous subsection, we can separately detect signals from different APs.

This simple approach is based on two observations.

1) Three non-overlapping WiFi channels: As shown in Fig. 3a, there are three non-overlapping WiFi channels: 1, 6, and 11. A network system manager tries to use these non-overlapping channels on each AP to minimize interference. We can assume that APs around a sensor node use different channels, which are typically non-overlapping channels.

Algorithm 1 Remove inseparable pairs of beacons

Require: a set of non-multiple beacon intervals L, $threshold$ for false detection rate.
Ensure: $L \Leftarrow$ a set of non-multiple beacon intervals

while sizeof$(L) > 1$ **do**
  all $inseparable[\,] = 0$
  for b_int in L **do**
    for $fold\_bint$ in $L \setminus \{b\_int\}$ **do**
      if $p(b\_int, fold\_bint) \geq threshold$ or $p(fold\_bint, b\_int) \geq threshold$ **then**
        $inseparable[b\_int] + +$
      end if
    end for
  end for
  if all $inseparable[\,] == 0$ **then**
    return L
  end if
  $unuse\_bint \Leftarrow b\_int$ associated with max$(inseparable)$
  $L \Leftarrow L \setminus \{unuse\_bint\}$
end while
return L

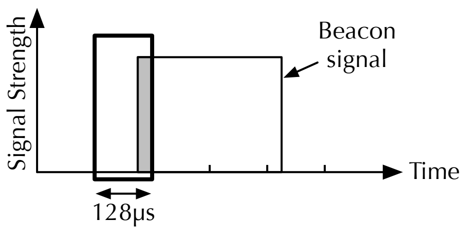

Fig. 5. Example of partial RSS problem. Only the gray part of a beacon signal contributes to RSS within a 128-microsecond window shown in the figure.

We need three APs for sensor localization using multilateration. Coincidentally, there are three non-overlapping channels that are typically used on APs. We therefore sample RSS on ZigBee channels 12, 17, and 22 that overlap WiFi channels 1, 6, and 11, respectively. If no AP is detected on ZigBee channels 12, 17, and 22, the other channels are used for RSS sampling.

2) Identical default beacon interval: Almost all APs available today are configured to use beacon interval of 100 TU, which is the de facto standard default configuration. On some APs, we cannot even change the beacon interval from 100 TU. We can easily detect APs by folding channel-usage samples by period $B = 100 \times 1024 \times 10^{-6}/(128 \times 10^{-6})$.

D. AP-RSS Extraction

After we detect AP beacon signals, we need to extract AP-RSS from the sequence of RSS samples. Although we can intuitively extract RSS samples corresponding to the columns whose superposition-rate exceeds a specific threshold, the extracted RSS samples are suffered from high RSS error. There are two main reasons for this RSS error.

-91	-53	-53	-53	-53	-53	-53	-53	-53	-53
-51	-51	-51	-52	-51	-51	-52	-61	-97	-79
-52	-52	-52	-52	-52	-52	-52	-52	-52	-52

Fig. 6. Example of RSS filtering in a partial RSS matrix. Black and gray boxes indicate that the channel is busy. We first remove RSS samples less than the channel-usage threshold (white boxes) and then cut off both the beginning and the end of the signal (gray boxes).

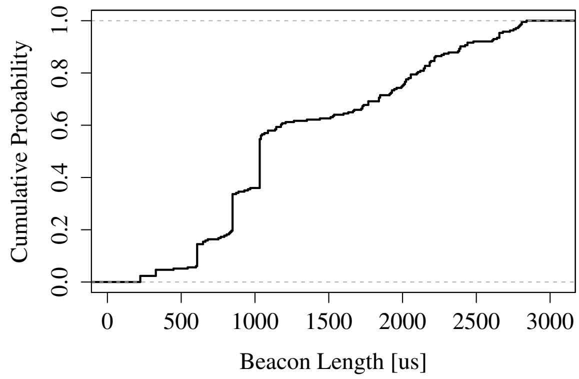

Fig. 7. Empirical cumulative distribution function of beacon length

1) *CSMA delay:*
   Due to the nature of a CSMA mechanism in WiFi MAC, beacon signals might be delayed. The intuitive RSS extraction picks RSS samples on scheduled timing, which results in the extraction of signals from other WiFi devices.
2) *Average RSS on a ZigBee module:*
   ZigBee modules can measure RSS averaged over 128 microseconds as defined in the standard. The ZigBee module might provide *partial RSS*, i.e., average RSS of a part of WiFi signal, as depicted in Fig. 5.

To minimize effects of the above two problems, we drop RSS on both rising edge and falling edge of each beacon signal. We first create an RSS matrix in a same manner as the creation of a channel-usage matrix. We then extract the columns corresponding to channel-usage matrix columns whose superposition-rate exceeds a specific threshold. Figure 6 shows an example of the extracted columns of an RSS matrix. We first remove RSS samples less than the channel-usage threshold defined in Section III-B (white boxes). We next remove the first and the last RSS samples on each row (gray boxes). These steps extract RSS samples of the core of beacon signals.

This simple filtering technique effectively works because most of the beacons have length more than four RSS-sample length, i.e., 512 microseconds. Figure 7 shows an empirical cumulative distribution function of beacon length of WiFi APs in our university building. More than 90 % of APs are sending beacons whose length is more than 512 microseconds.

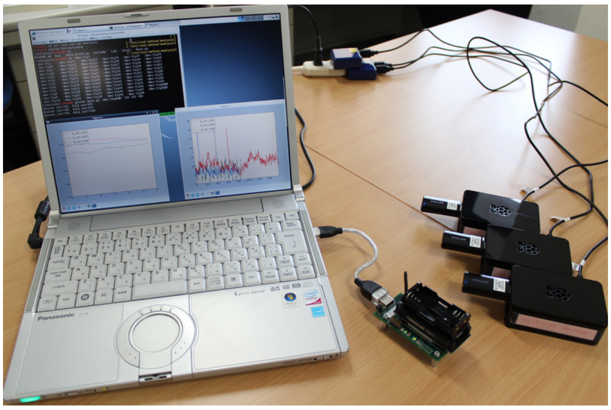

Fig. 8. AP-RSS monitoring system

IV. Implementation

To conduct experimental evaluations, we implemented an AP-RSS monitoring system using off-the-shelf devices. Figure 8 shows our AP-RSS monitoring system. The AP-RSS monitoring system could reveal RSS of three APs simultaneously.

We used a Raspberry Pi B+ employing a WiFi module WLI-UC-G301N from Buffalo as a WiFi AP. OpenWrt, an open source OS for WiFi APs, was running on Raspberry Pi.

We used a MICAz sensor node from Crossbow that utilizes a CC2420 IEEE 802.15.4 module from Texas Instruments. We developed a C program that samples RSS every 128 microseconds and sends the RSS to a laptop via serial communication interface.

The laptop was CF-Y8 from Panasonic. We developed a python program that receives RSS from the sensor node and extracts AP-RSS as described in Section III.

V. Evaluation

To demonstrate the feasibility of the sensor localization system described in Section II, we evaluated the detection rate, false detection rate, and RSS error of our RSS monitoring system. We first conducted a preliminary experiment to determine the number of folding and a superposition-rate threshold. We then evaluated the detection rate, false detection rate, and RSS error.

A. Preliminary Experiment

We conducted a preliminary experiment to determine the number of folding and a superposition-rate threshold, as described in Section III-B. We installed a WiFi AP 2.5-meter away from a sensor node that was connected to a laptop. We then collected a sequence of RSS for one minute. The beacon interval of the AP was fixed to 109 TU. The WiFi AP and sensor channels were configured to 11 and 22, respectively. These channels overlap as shown in Fig. 3a.

Figure 9 shows average superposition-rate as a function of the number of folding N_F; error bars indicate minimum and maximum superposition-rate. Figure 9 shows the following:

1) Average superposition-rate were always less than 100 %. There were beacon transmission delays due to a CSMA/CA mechanism.
2) Increase in the number of folding N_F resulted in small fluctuations of superposition-rate. When $N_F \geq 30$, the variance of superposition-rate was less than 15 %. When N_F was small, change in a channel-usage sum had a significant effect on superposition-rate.
3) Increase in N_F resulted in decrease in average superposition-rate. The beacon interval and RSS sampling period were not exactly equal. Sampling period error accumulated as N_F increased and some beacon signals appeared in a different column in a channel-usage matrix.

Using the above results, we determined the number of folding N_F as 30 to retrieve a stable and high superposition-rate. When $N_F = 30$, superposition-rate was always greater than 90 %. We therefore used a superposition-rate threshold of 80 %, which includes threshold margin of 10 %.

B. Detection Rate and False Detection Rate

We evaluated a detection rate and false detection rate to confirm that our AP-RSS monitoring system can recognize sender APs. A detection rate is the probability that beacon signals of interval t are detected as interval t. A false detection rate is the probability that beacon signals whose interval is not t are detected as interval t signals.

We generated a set of separable beacon intervals as described in Section III-C. The threshold of false detection rate in algorithm 1 was 10 %. Beacon intervals were selected from 15 to 500 TU, which are practically used on actual APs. A WiFi AP was installed 2.5-meter away from a sensor node connected to a laptop that collected RSS samples. For each beacon interval, we configured the AP to have the interval and sampled RSS on the sensor for one minute. Using the RSS samples, we calculated the detection rate and false detection rate. The false detection rate of interval t had multiple values

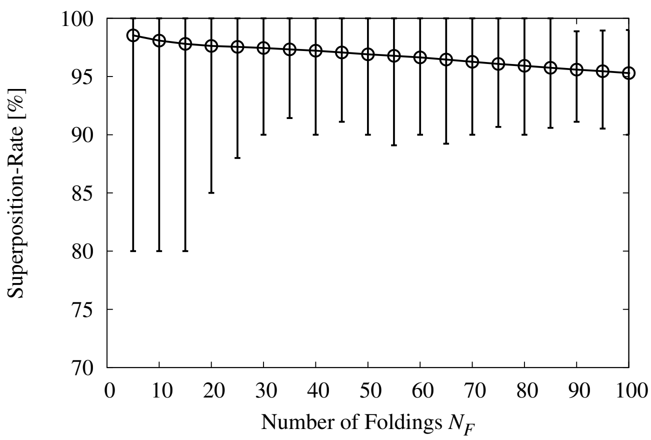

Fig. 9. Superposition-rate as a function of the number of folding N_F; error bars indicate minimum and maximum

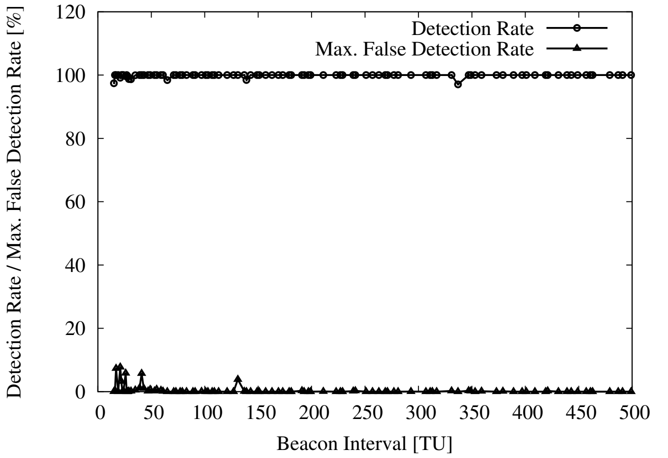

Fig. 10. Detection rate and maximum false detection rate as a function of beacon interval

because there were multiple beacon intervals that is not t. We therefore evaluated the maximum value of the false detection rate for each beacon interval t.

Figure 10 shows the detection rate and maximum false detection rate as a function of beacon interval. Figure 10 shows the following:

1) The detection rate was greater than 95 % for all the beacon intervals. The minimum detection rate was 97.1 % when beacon interval was 337 TU.
2) The maximum false detection rate was less than 10 % for all the beacon intervals. The maximum false detection rate was 7.76 % when beacon interval was 21 TU.

The above results demonstrated that our AP-RSS monitoring system could successfully recognize beacon signals with a probability of false detection less than 10 %.

C. RSS Error

To calculate RSS error, we compared AP-RSS with true RSS. The AP-RSS is RSS derived by our system and the true RSS is the RSS directly derived from a WiFi module. We first configured a WiFi AP to have beacon interval of 93 TU and installed the AP 12-meter away from a sensor node connected to a laptop. We then collected a sequence of AP-RSS for four seconds using our AP-RSS monitoring system. At the same time, we measured true RSS by capturing beacon signals on the laptop using a libpcap library. The AP-RSS and the true RSS are averaged over four seconds. We collected the averaged four-second AP-RSS as well as true RSS for 1,000 times.

Figure 11 shows AP-RSS and true RSS of each trial. We revealed the following:

1) The AP-RSS exhibited similar fluctuation pattern to that of the true RSS.
2) There was an offset between the AP-RSS and true RSS. This offset was due to the bandwidth difference between WiFi and ZigBee: WiFi bandwidth is 22 MHz, while ZigBee bandwidth is 2 MHz. Antenna gain was another source of the offset.

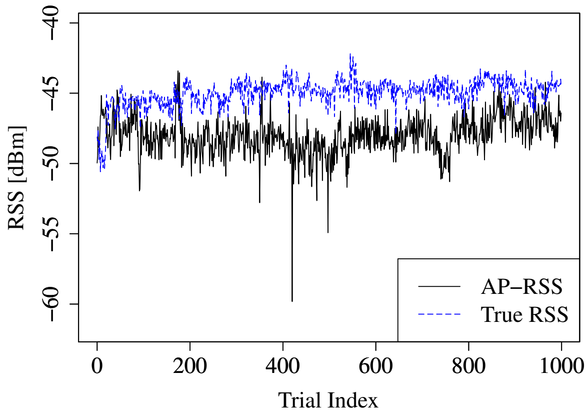

Fig. 11. AP-RSS and true RSS. The AP-RSS is RSS derived by our AP-RSS monitoring system. The true RSS is RSS derived from a WiFi module on a laptop.

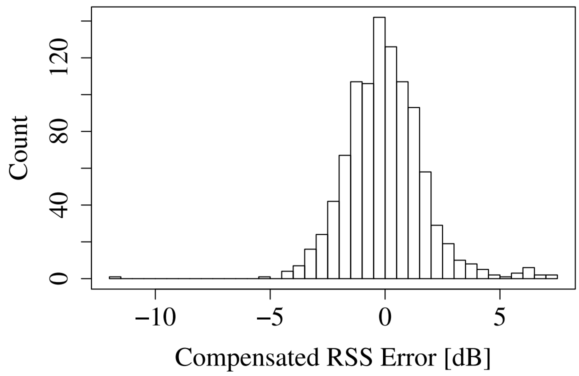

Fig. 12. Histogram of compensated RSS error

For RSS error calculation, we compensated the offset because the effect of bandwidth difference and antenna gain were almost constant. The offset was calculated as difference between the average AP-RSS and average true RSS over 1,000 times.

Figure 12 shows a histogram of compensated RSS error. The RSS error followed Gaussian distribution. The standard deviation of compensated RSS error was 1.71 dB and the average absolute error was 1.26 dB. More than 95 % of errors were within $2 \times \pm 1.71$ dB $= \pm 3.42$ dB. We can conclude that the RSS error was at the same order of RSS fluctuation due to the environmental change.

VI. Related Works

In the field of indoor localization, previous studies have primarily investigated reduction in deployment costs and accuracy improvement. Most of these works are using WiFi devices, which still can be applied to sensor nodes with ZigBee modules.

Iterative multilateration [4, 5] uses localized nodes as new anchor nodes, which reduces the number of initial anchor nodes. However, many initial anchors are still required to achieve small localization error in a large building. Crowdsourcing combined with fingerprinting localization [6–13] is another approach which reduces deployment costs. For a sensor localization system, it is difficult to get user cooperation because almost all users are carrying no ZigBee devices.

In this paper, we focus on cross-technology RSS extraction since we can employ existing localization method using RSS. Previous works on accuracy improvement [14–20] is therefore useful in our future work, i.e., localization using the extracted RSS.

There is a new fingerprinting localization named ZiFind which requires no anchor nodes [27]. ZiFind, however, requires many WiFi devices called ZiFind mappers instead of anchor nodes. Cross-technology communication have also been studied [28–30], which requires special hardwares or firmware modification on WiFi APs or sensor devices.

VII. Conclusion

In this paper, we present an AP-RSS monitoring system using sensor node, as a first step toward a sensor localization system using WiFi APs as anchors. We developed a cross-technology signal extraction scheme to overcome the wireless technology difference between WiFi and ZigBee. In our signal extraction scheme, a signal processing technique presented in an existing work is employed and combined with a simple filtering method to extract AP-RSS on sensor nodes. We also present a simple AP recognition technique that uses different beacon intervals with some constraints on each AP. The experimental evaluations showed that our AP-RSS monitoring system successfully retrieved AP-RSS with an average error of 1.26 dB while recognizing sender APs.

Acknowledgment

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

References

[1] J. Wang, R. K. Ghosh, and S. K. Das, "A survey on sensor localization," *J. Control Theory Applications*, vol. 8, no. 1, pp. 2–11, Feb. 2010.

[2] L. Cheng, C. Wu, Y. Zhang *et al.*, "A survey of localization in wireless sensor network," *Int. J. Distributed Sensor Networks*, vol. 2012, pp. 1–12, article ID 962523, Nov. 2012.

[3] A. Lédeczi and M. Maróti, "Wireless sensor node localization," *Philosophical Trans. Royal Society A*, vol. 2012, no. 370, pp. 85–99, Jan. 2012.

[4] M. Minami, Y. Fukuju, K. Hirasawa *et al.*, "DOLPHIN: A practical approach for implementing a fully distributed indoor ultrasonic positioning system," in *LNCS*, vol. 3205, pp. 437–365, Sep. 2004, Proc. ACM Ubicomp.

[5] L. Huang, F. Wang, C. Ma *et al.*, "The analysis of anchor placement for self-localization algorithm in wireless sensor networks," in *Advances Wireless Sensor Networks, Communications in Computer and Info. Science*, vol. 334, pp. 117–126, 2013.

[6] P. Bolliger, "Redpin – adaptive, zero-configuration indoor localization through user collaboration," in *Proc. ACM Int. Workshop Mobile Entity Localization Tracking GPS-less Environments (MELT)*, pp. 55–60, Sep. 2008.

[7] A. Barry, B. Fisher, and M. L. Chang, "A long-duration study of user-trained 802.11 localization," in *LNCS*, vol. 5801, pp. 197–212, Sep.–Oct. 2009, Proc. ACM Int. Workshop Mobile Entity Localization Tracking GPS-less Environments (MELT).

[8] J.-G. Park, B. Charrow, D. Curtis *et al.*, "Growing an organic indoor location system," in *Proc. ACM MobiSys*, pp. 271–284, Jun. 2010.

[9] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan *et al.*, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM MobiCom*, pp. 293–304, Aug. 2012.

[10] H. Wang, S. Sen, A. Elgohary *et al.*, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM MobiSys*, pp. 197–210, Jun. 2012.

[11] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM MobiCom*, pp. 269–280, Aug. 2012.

[12] C. Wu, Z. Yang, Y. Liu *et al.*, "WILL: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 839–848, Apr. 2013.

[13] Z. Jiang, J. Zhao, J. Han *et al.*, "Wi-Fi fingerprint based indoor localization without indoor space measurement," in *Proc. IEEE Int. Conf. Mobile Ad-Hoc and Sensor Systems (MASS)*, pp. 384–392, Oct. 2013.

[14] A. Taok, N. Kandil, and S. Affes, "Neural networks for fingerprinting-based indoor localization using ultra-wideband," *J. Communications*, vol. 4, no. 4, pp. 267–275, May 2009.

[15] G. S. Kuruoglu, M. Erol, and S. Oktug, "Three dimensional localization in wireless sensor networks using the adapted multi-lateration technique considering range measurement errors," in *Proc. IEEE GLOBECOM Workshops*, pp. 1–5, Nov.–Dec. 2009.

[16] A. W. Tsui, Y.-H. Chuang, and H.-H. Chu, "Unsupervised learning for solving RSS hardware variance problem in WiFi localization," *Mobile Networks and Applications*, vol. 12, no. 5, pp. 677–691, Oct. 2009.

[17] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Intelligent dynamic radio tracking in indoor wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 405–419, Mar. 2010.

[18] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLAN's received signal strength indication for indoor location fingerprinting," *Pervasive and Mobile Computing*, vol. 8, no. 2, pp. 292–316, Apr. 2012.

[19] S. Sen, B. Radunović, R. R. Choudhury *et al.*, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proc. ACM MobiSys*, pp. 183–196, Jun. 2012.

[20] N. Wirström, P. Misra, and T. Voigt, "Spray: A multi-modal localization system for stationary sensor network deployment," in *Proc. Annual Conf. Wireless On-demand Network Systems Services (WONS)*, pp. 25–32, Apr. 2014.

[21] R. Zhou, Y. Xiong, G. Xing *et al.*, "ZiFi: wireless LAN discovery via ZigBee interference signatures," in *Proc. ACM MobiCom*, pp. 49–60, Sep. 2010.

[22] S. Ishida, K. Izumi, S. Tagashira *et al.*, "WiFi AP-RSS monitoring using sensor nodes toward anchor-free sensor localization," in *Proc. IEEE Vehicular Technology Conf. (VTC-Fall)*, pp. 1–5, Sep. 2015.

[23] IEEE Standards Association, "IEEE Std 802.15.4-2011, IEEE standard for local and metropolitan area networks — part 15.4: Low-rate wireless personal area networks (LR-WPANs)," Sep. 2011, http://standards.ieee.org/.

[24] Texas Instruments, "CC2420: Single-chip 2.4 GHz IEEE 802.15.4 compliant and ZigBee ready RF transceiver," datasheet, http://www.ti.com/.

[25] S. M. Kim and T. He, "FreeBee: Cross-technology communication via free side-channel," in *Proc. ACM MobiCom*, pp. 317–330, Sep. 2015.

[26] IEEE Standards Association, "IEEE Std 802.11-2007, IEEE standard for local and metropolitan area networks — part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," Jun. 2007, http://standards.ieee.org/.

[27] Y. Gao, J. Niu, R. Zhou *et al.*, "ZiFind: Exploiting cross-technology interference signatures for energy-efficient indoor localization," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, pp. 2940–2948, Apr. 2013.

[28] K. Chebrolu and A. Dhekne, "Esense: Energy sensing-based cross-technology communication," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2303–2316, Nov. 2013.

[29] X. Zhang and K. G. Shin, "Gap Sense: Lightweight coordination of heterogeneous wireless devices," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, pp. 3094–3101, Apr. 2013.

[30] Y. Kondo, H. Yomo, S. Tang *et al.*, "Wake-up radio using IEEE 802.11 frame length modulation for radio-on-demand wireless LAN," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 869–873, Sep. 2011.