WiBee: Building WiFi Radio Map with ZigBee Sensor Networks

Wenxian Li[1], Yanmin Zhu[1,2,*], Tian He[3]
[1] CSE Department, Shanghai Jiao Tong University; [2] Shanghai Key Lab of Scalable Computing and Systems;
[3] University of Minnesota; [*] Corresponding Author
{wofanli, yzhu}@sjtu.edu.cn; tianhe@cs.umn.edu

Abstract—Exploiting the increasing ubiquitous deployment of sensor networks, the paper presents a system called WiBee that utilizes ZigBee sensors to build real-time WiFi radio maps. The design of WiBee is motivated by the observation that a ZigBee radio can sense WiFi frame transmissions although it cannot decode WiFi frames. A sensor passively listens on the wireless channel and estimates the RSS of a WiFi AP at its location. The design of WiBee faces three unique challenges. First, multiple APs may transmit frames concurrently and frame collisions may happen. Second, because of severe resource constraints, a sensor cannot sample the channel at arbitrarily high frequency and hence some frame transmissions may not be sampled. Third, sensor nodes are usually not time synchronized and the on-board clock is inaccurate. To address these challenges, we propose a novel gateway-assisted approach to estimating WiFi RSS values at ZigBee sensors. A light-weight algorithm is designed for identifying the RSS values corresponding to a given AP. It searches the sequence of ZigBee RSS samples for an AP signature sequence. An optimization technique is proposed to address issues of clock drift and time asynchronization. Our extensive experiments on a testbed show that WiBee can achieve low estimation error, short delay and small computation overhead.

Keywords—WiFi, ZigBee, Sensor Network, Radio Map, Signal Strength, RSS

I. INTRODUCTION

A WiFi radio map shows received signal strength (RSS) of WiFi access points (APs) at different locations in a given environment. Such a radio map is useful to many applications such as AP selection and localization [11]. With a radio map, a user can select the AP that transmits stronger RSS or even move to a location with better RSS. As a result, the user can enjoy a connection with better quality.

A direct approach to building a WiFi radio map is to have a mobile device with a WiFi network interface card (NIC) sweep across the given space, scanning all channels for RSS at each location [11]. However, this approach is limited due to its time consuming way of sweeping the entire space.

In this paper we present a novel system called WiBee for building real-time WiFi radio maps with a ZigBee sensor network. The system exploits the increasing deployment of sensor networks in various sectors, such as average household and offices [7, 9]. The design of WiBee is motivated by the observation that a ZigBee radio can sense WiFi signals although it cannot decode WiFi frames. Our measurement study shows that there is steady relation between RSS samples read by ZigBee radios and RSS samples read by WiFi radios.

However, the design of WiBee faces three unique challenges: 1) Multiple APs and clients may transmit frames concurrently. Since a sensor cannot decode the frames, it is difficult for a sensor to associate RSS samples with a specific AP. In addition, frame collisions may happen. As a result, the signal sensed by a sensor may be a mixture of several AP signals and it is difficult for a sensor to distinguish mixed signals. 2) Because of severe resource constraints on power, energy and memory, a sensor cannot sample the channel at arbitrarily high frequency. When reading from the RSS indicator (RSSI) register at a high frequency, the processor for the mote may suffer high utilization and leave other tasks unable to access the processor. Furthermore, a large set of samples should be stored locally but the storage on a sensor node is very limited. 3) Sensor nodes are usually not time synchronized. In addition, the on-board clock on a sensor is inaccurate and associated with an unknown clock drift. Considering the sampling nature of RSS values, strict timing is important for identifying the RSS values for a given AP. Unknown clock drift makes strict timing difficult.

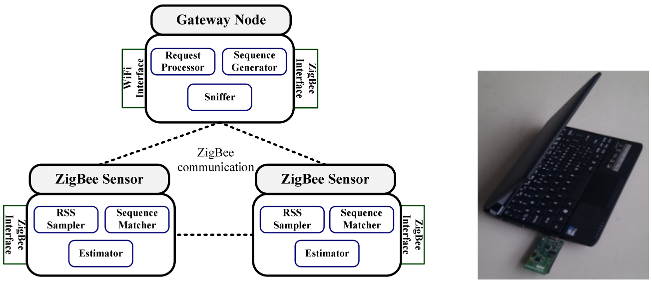
To tackle these challenges, we propose a novel gateway-assisted approach for estimating WiFi RSS of APs at sensors. A new light-weighted algorithm is designed for a sensor to identify the RSS values corresponding to a given AP. The central idea is to search the sequence of ZigBee RSS samples for an AP signature sequence generated from the sequence of AP frames. An optimization technique is proposed to address issues of unknown clock drift and time asynchronization. We have implemented WiBee with Linux laptop gateway and TelosB motes. Our experiments show that WiBee can estimate WiFi RSS with high accuracy and short delay. We envision that WiBee will become increasingly useful for wireless environments where users want to understand WiFi RSS at different locations.

We have made the following contributions in this paper. First, we make the first attempt to building a real-time WiFi radio map with a ZigBee sensor network. Each sensor merely listens on the channel and reads RSS samples with its ZigBee radio. Second, we design a novel gateway-assisted approach that allows resource-constraint sensors to estimate the WiFi RSS of an AP. We propose a matching algorithm that indentifies the subsequence of RSS values that correspond to a given AP. Finally, we have implemented WiBee with a Linux laptop gateway and TelosB motes. Our extensive experiments on a testbed consisting of two emulated APs, Linux laptop gateway and 9 TelosB motes show that WiBee can achieve low estimation error, short delay and small computation overhead.

II. DESIGN OF WIBEE

The main steps for WiBee to create a radio map are described as follows.

Step 1: A WiFi client sends a map request to the gateway



(a) (b)

Figure 1: (a) The architecture of the WiBee system; (b) An example gateway node that is the combination of a netbook (eMachine 350) and a TelosB mote connected via the USB interface.

through WiFi, and the gateway broadcasts the request to the sensors through ZigBee.

Step 2: On receiving the request, each sensor begins to listen on the specified channel and read RSS samples.

Step 3: The gateway sniffers on the channel specified by the request, and captures a sequence of WiFi frames. A sequence of frame digests of each AP is generated, which is then sent to all sensors.

Step 4: Each sensor estimates the RSS of each APat the sensor's location. WiFi RSS estimates are routed back to the gateway.

Step 5: The gateway collects all RSS estimates from the sensors, creates the map and sends this map to the WiFi client.

In the following, we discuss the major design details of WiBee according to Figure 1.

A. *Sequence Generator at Gateway*

The gateway tries to capture all frames. It then extracts the sequence of the frames belonging to each different AP. For each frame, the gateway creates a digest that includes length d and inter-arrival time to the next frame, denoted as Δt, as shown in Figure 3.

B. *RSS Sampling at Sensor*

Each sensor periodically senses the channel and read RSS samples. The main design issues concerning RSS sampling are sampling frequency, and sampling window size. Let I denote the sampling interval. The sampling window size W_S determines the duration that the sensor reads RSS values. However, as the inaccurate clock, the widow size W_S'measured locally by the sensor is different from the real W_S [5, 12]. According to [12], the clock drift rate is approximately constant in a short time window. Denote the drift rate as ψ. Thus, we have,

$$W_S = \Psi W_S'. \tag{1}$$

C. *Taming Time Asynchronization*

The gateway and sensors are not time synchronized. WiBee solves the issue of time asynchronization as follows. As mentioned, it takes time τ_{delay} for the network to deliver the request to this senor. Thus, in WiBee the gateway postpones for time τ_{guard} to capture the frames. The length τ_{guard} should accommodate the maximum τ_{delay} in the sensors, i.e., $t_a > t_c$ as shown in Figure 2.

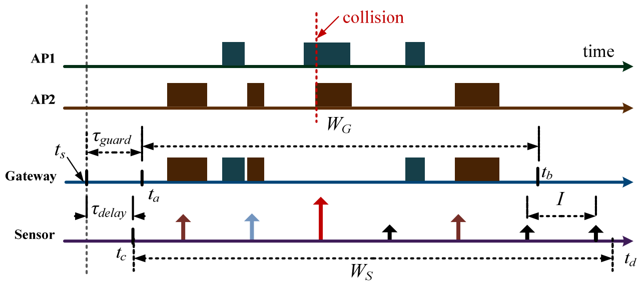


Figure 2: Timing illustration of frames and RSS readings. The sensor reads RSS samples. Some RSS values correspond to no received frames at the gateway because of frame collisions.

D. *Light-weight Sequence Matching Algorithm*

After having the sequence of frame digests of an AP and the RSS samples, the sensor next needs to identify the RSS values that correspond to this AP. The key step in this system is to divide the complete sequence of the RSS samples, denoted as X, into several subsequences according to the WiFi frame digest sequence. Each subsequence is then assigned to a particular AP.

1) *Shaping Frame Digest and RSS*

As the computational ability of a simple sensor is very limited, we first shape both the digest and the RSS sequences. For easy of computation, the time is divided into slots of I. The sensor adjusts each RSS sample according to two criteria: 1) The magnitude of an RSS sample is set to zero if it is below -90dBm. 2) All the remaining RSS samples are set to 1 to indicate the busy channel. Let the sequence of the original RSS samples is denoted by $X = (x_1, x_2, \dots, x_n)$, and the sequence of shaped RSS samples is denoted by $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$, where n is the number of RSS samples, $n = W_S/I$.

We next shape the sequence of frame digests and create a signature sequence of 0 or 1 values, 1 as busy channel and 0 as idle channel. Let the created sequence denoted by $\tilde{Y} = (\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_\varepsilon)$, where $\varepsilon = W_G/I$. The values in the sequence is determined by,

$$\tilde{y}_i = \begin{cases} 1, \text{if } t = t_a + \tau + i \cdot I \text{ contained in a frame} \\ 0, \text{otherwise} \end{cases}. \tag{2}$$

where τ is a parameter larger than zero. This parameter is introduced to accommodate the misalignment between the gateway and the sensor.

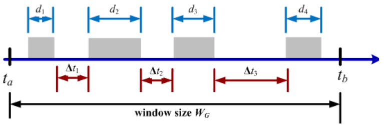


Figure 3: The generated sequence of frame digests of anAP.

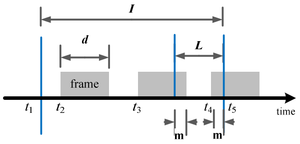
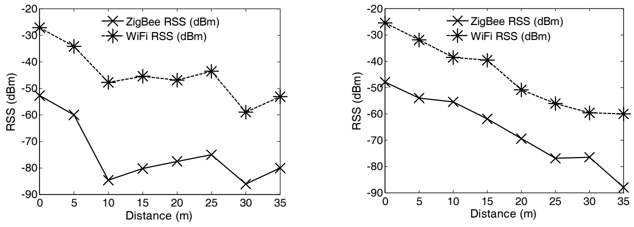


Figure 4: ZigBee RSS sampling model.



(a) indoor (b) outdoor

Figure 5: Comparison of RSS by ZigBee mote and RSS by 802.11 laptop (indoor and outdoor).

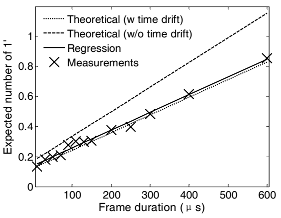


Figure 6: Analytical results and real measurements on the expected number of 1's.

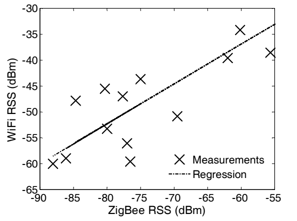


Figure 7: The real measurements and the model curve fitted with linear least squares regression.

2) Sequence Matching

After deriving the shaped RSS sequence and the AP signature sequence, we need to identify the RSS values of the given AP by aligning the two sequences. There are three main issues: 1) The two sequences may start at different starting points because of the uncertain delay introduced by request delivery from the gateway to the sensor; 2) The on-board clock of the sensor is inaccurate, and there is an unknown clock drift rate; 3) RSS values may result from not only frame transmissions but also radio interferences and noises.

To tackle the challenges, we propose a novel algorithm for aligning the two sequences and hence identifying the RSS values for the AP. It searches the RSS sequence $\tilde{X}$ for the signature sequence $\tilde{Y}$. Since the RSS values are affected by transmission interference and noises, it is hardly possible that there is an exact match between $\tilde{Y}$ and a subsequence of $\tilde{X}$. However, it is still reasonable to believe there is such a subsequence in $\tilde{X}$ that shares much similarity with $\tilde{Y}$. This RSS subsequence corresponds to the frame transmissions of the AP producing $\tilde{Y}$. Here, we consider a subsequence $\tilde{X}_{sub} = (s_1, s_2, \ldots, s_\varepsilon)$ in $\tilde{X}$ that aligns with $\tilde{Y}$. Suppose $\tilde{Y}$ aligned to the ith position of $\tilde{X}$. Then, we construct $\tilde{X}_{sub}$,

$$s_j = \begin{cases} \tilde{x}_{i-1+j}, \text{if } \tilde{y}_j = 1 \\ 0, \text{if } \tilde{y}_j = 0 \end{cases}, 0 \le j \le \varepsilon \tag{3}$$

To measure the similarity between two sequences, we employ Pearson correlation [8], as follows,

$$corr(\tilde{X}_{sub}, \tilde{Y}) = \frac{\text{cov}^2(\tilde{X}_{sub}, \tilde{Y})}{\text{cov}(\tilde{X}_{sub}, \tilde{X}_{sub})\,\text{cov}(\tilde{Y}, \tilde{Y})} \tag{4}$$

To account for the uncertain delay for request delivery, we introduced τ in (2). To search for the signature sequence in the RSS sequence, τ should be searched from 0 to τ_{delay}. Each tried time we increment τ by a unit of a symbol time (i.e., 16μs).

One remaining issue is the clock drift. To account for the clock drift on the sensor, we should modify (2) by replacing I with I', where I' is the sampling interval measured with the sensor's local clock,

$$I' \times \psi = I. \tag{5}$$

We also have a very important observation that the RSS values from the same AP should be stable, i.e., the variance of these values is small. To exploit this observation, we propose a new metric λ for evaluating a match,

$$\begin{aligned} \lambda &= corr(\widetilde{X}_{sub}, \widetilde{Y}) + w \times \frac{1}{\delta(X_{sub})} \\ \delta(X_{sub}) &= \sum_{i=1}^{\varepsilon} (x_i - m(X_{sub}))^2 \end{aligned} \tag{6}$$

where w is a weight coefficient, $w \in [0, 1]$.

Finally, the search problem can be formulated as an optimized problem as shown below:

$$\begin{aligned} \max \quad & \lambda(\tau, \psi) \\ \text{s.t.,} \quad & \tau \in [0, \tau_{delay}] \\ & \psi \in [\psi_l, \psi_h] \end{aligned} \tag{7}$$

where ψ_l and ψ_h are the lowest and the highest clock drift rate, respectively.

3) Differentiating Frame Importance

WiFi frames with different length have different degrees of importance. A longer frame is more important because it has a higher probability of being sampled by a sensor. We characterize the importance of a frame by the expected number of ones in the shaped RSS sequence. Let this expected number be denoted by $\hat{v}$. The larger $\hat{v}$ is, the more important the corresponding frame is.

We next derive the expected number v. As shown in Figure 4, when a frame reaches at t_2, it cannot be sampled at t_1 or t_5. We assume the arrival time of frame obeys uniform distribution. Then, v can be computed as,

$$v_i = \left| \frac{d_i}{I} \right| + \frac{t_4 - t_3}{I} = \frac{d_i}{I} + \frac{L}{I}\left(1 - \frac{2T}{R_i - N}\right) \tag{8}$$

where R is the real RSS of the WiFi frame, N is the noise floor of the channel, and T is the RSS difference threshold between busy and idle channel.

We verify the analysis with real experiments, as shown in Figure 6. Here, we have the following observation. Without consideration of the clock drift, the theoretical line does not match the curve measured in the experiments. Time drift is large enough to affect the results.

E. WiFi RSS Estimation

After identifying the RSS values that correspond to the AP, the sensor next estimates the RSS of the AP based on these identified RSS values.

First, we estimate the ZigBee RSS value. Let $X^*_{sub} =$

$(x_i, x_{i+1}, \dots, x_{i+\varepsilon-1})$ denote the subsequence corresponding to the matched $\tilde{X}^*_{sub}$. Each RSS value $x_j, i \le j \le i+\varepsilon-1$ corresponds to a WiFi frame and let v_j denote the importance degree of this frame. Thus, we estimate the RSS with a weighted average,

$$R_{sensor} = \sum_{j=1}^{\varepsilon} x_{i+j-1} \times s_j \times v_{i+j-1} \Big/ \sum_{j=1}^{\varepsilon} v_{i+j-1} . \qquad (9)$$

Next, we estimate the RSS of the AP, denoted by RSS_{WiFi}. As shown by our empirical study in Figure 5, the difference of the ZigBee RSS and the WiFi RSS is almost a constant. Based on this observation, we propose a linear model,

$$RSS_{wifi} = \alpha + \beta \times RSS_{zigbee} \qquad (10)$$

where α and β are two model parameters. Noted that α and β are platform dependent. To estimate the two parameters, we utilize the technique of Least-Squares Regression [2]. The fitting result based on the real measurements is shown in Figure 7.

III. EXPERIMENTATION

We have implemented WiBee with a Samsung laptop operating the Fedora 14 Linux operating system and TelosB motes operating TinyOS2.

We investigate two performance metrics, i.e., error rate and delay. The error rate ϖ is defined as follows.

$$\varpi = \left| \left(RSS_{wifi} - RSS_{actual} \right) / RSS_{actual} \right| \qquad (11)$$

where RSS_{wifi} is the estimated RSS by a mote and RSS_{actual} is the real RSS. The delay for building a map is the time from the instant a request is received until the instant a radio map is received.

A. Experimental Setup

As shown in Figure 8, a testbed is setup for experiments, which consists of a laptop gateway (Samsung R18), two netbooks (eMachine 350) with an 802.11b NIC and nine TelosB motes. The nine motes form a 3×3 grid. The grid size is 5m×5m.

One eMachine netbook is used to emulate a target WiFi AP. The other is used to emulate a coexisting AP that generates different frame traffic. We call the later an interference AP. The advantage of using an emulated AP is that we can control the generation of WiFi traffic. Traffic of both the APs is generated using high-precision Internet traffic generator D-ITG [1].

Each TelosB mote is running TinyOS 2. Each mote is one hop connected to the mote on the gateway. The transmit power of each sensor is set to 0 dBm. If not specified elsewhere, the window size is 300 ms, τ_{guard} is 10 ms, and the sampling interval at a mote is 600 µs. It should be noted that RSS_{actual} is measured at the mote's location at a later time by a WiFi netbook.

B. Experimental Results

We first investigate the accuracy of WiFi RSS estimation in terms of error rate and FN rate. As the channel utilization has a considerable impact on estimation accuracy, we study accuracy as channel utilization of the interference AP is varied.

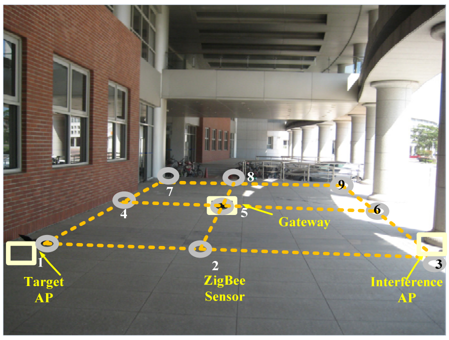


Figure 8: The experimental setup at the open corridor of an academic building. The nine sensors form a 3×3 grid (5m separation). There is a gateway at the center, an interference AP at the coroner and a request client.

In Figure 9, we show estimation error rate under different channel utilization of the interference AP. We can find that the error rate is low, dependent on the channel initialization of the interference AP. In Figure 10, we show error rates for individual sensors over different experiments. In Figure 11, a radio map consisting RSS values for the locations of 9 sensors is shown. Note that the target AP is close to node 3. We can find that the RSS at the location of node 3 is strongest as it is closest to the target AP. We also study the FN rate achieved by WiBee. Figure 12 shows FN rate for different channel utilization. The FN rate is steadily below 10%.

We next investigate the effect of differentiating the importance of frames with different frame lengths. The simple average scheme estimates WiFi RSS with an arithmetic average is used as the comparison. Figure 13 shows the effectiveness of the proposed differentiation mechanism.

We next study the delay required for requesting a radio map from this testbed. Two parameters influence the delay, including gateway window size and τ_{guard}. In Figure 14, delay is shown for different numbers of frames. In general, the delay increases as the number of frames in the window increases. In Figure 15, delay is plotted for different τ_{guard}. We can see the delay is 9 seconds when τ_{guard} is around 10ms. In general, the delay increases as τ_{guard} increases. In a typical WiFi environment, τ_{guard} is usually smaller than 10 ms. Figure 16 shows how to select a proper ρ considering the tradeoff between FN rate and error rate.

IV. RELATED WORK

To build radio maps, most previous techniques [11] sweep over a given space and store average RSS values. It takes a long time before a radio map can be constructed. In addition, it is difficult for this method to update the radio map. Radio interference [10] is an important issue when building radio maps.

Sensor networks [4] are becoming more and more popular and increasingly deployed. Building an indoor radio map with many sensors has been studied. In [6], indoor radio maps are

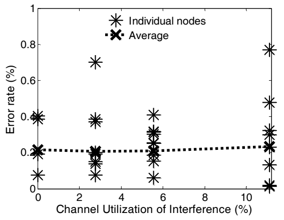


Figure 9: Estimation error rate vs. channel utilization of interference.

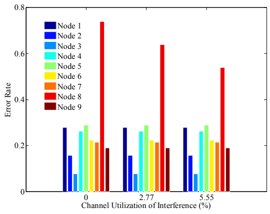


Figure 10: Error rates for individual sensors over different experiments.

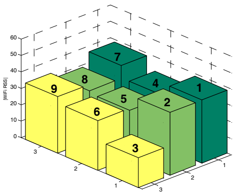


Figure 11: A radio map consisting of RSS values at 9 locations (absolute RSS values).

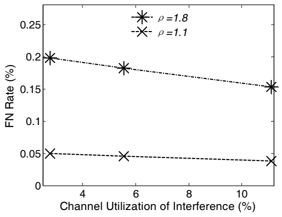


Figure 12: FN rates vs. channel utilization.

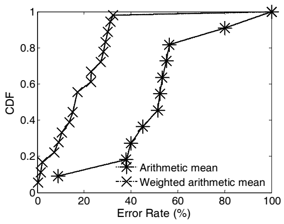


Figure 13: CDFs of error rates for simple average and weighted average.

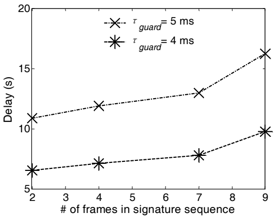


Figure 14: Delay vs. number of WiFi frames.

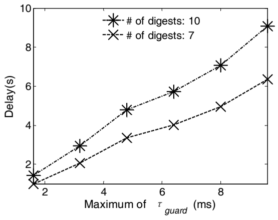


Figure 15: Delay vs. τ_{guard}.

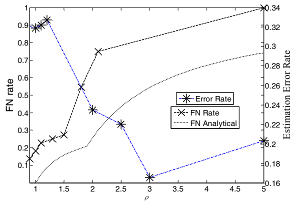


Figure 16: FN rate and error rate vs. ρ (two y-axis's, one FN rate and the other error rate).

constructed with a testbed of spectrum sensors with different measurement capabilities. Its focus is on the study of the indoor radio propagation conditions of a single sensor node. Distinct from that work, our paper focuses on building a radio map for each WiFi AP in the environment with ZigBee sensors.

Some pioneering studies have recognized the co-existence of WiFi and ZigBee [3, 13]. In [13], the authors develop a system called ZiFi to detect availability of WiFi APs via ZigBee radios. It searches for periodic signals that are caused by periodic beacons often used in 802.11 protocols. ZiFi is relevant to ours. As pointed out by the authors, however, many commercial WiFi NICs use similar beacon periods. In this case, ZiFi could not distinguish these APs. As a consequence, ZiFi is not able to estimate the RSS of a given AP. In contrast, WiBee works even if APs share the same beacon period.

V. CONCLUSION

In this work we have developed a system called WiBee for building WiFi radio maps with ZigBee sensor nodes. A light-weight algorithm has been designed for resource-constrained sensors to identify the RSS values that correspond to a WiFi AP and then the WiFi RSS at the location of the sensor is estimated. The algorithm deals with several unique challenges including frame interference, time unsynchronization, and clock drift. We have implemented WiBee with a Linux laptop gateway and TelosB motes. Our experimental results show that WiBee can estimate WiFi RSS with low estimation error and short delay.

ACKNOWLEDGMENT

This research is supported by Shanghai Pu Jiang Talents Program (10PJ1405800), Shanghai Chen Guang Program (10CG11), NSFC (No. 61170238, 60903190, 61027009, 60970106, 61170237), 973 Program (2005CB321901), MIIT of China (2009ZX03006-001-01), Doctoral Fund of Ministry of Education of China (20100073120021), 863 Program (2009AA012201 and 2011AA010500), and HP IRP (CW267311). In addition, it is partially supported by the Open Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2010KF-04), Beijing University of Aeronautics and Astronautics.

REFERENCES

[1] A. Botta, A. Dainotti, and A. Pescape, "Multi-protocol and Multi-platform Traffic Generation and easurement, Demo," *Proc. IEEE INFOCOM*, 2007.
[2] J. W. Demmel, in *Applied Numerical Linear Algebra*, 1997.
[3] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond Co-existence: Exploiting WiFi White Space for ZigBee Performance Assurance," *Proc. IEEE ICNP*, 2010.
[4] M. Li and Y. Liu, "Rendered Path: Range-Free Localization in Anisotropic Sensor Networks with Holes," *IEEE/ACM Transactions on Networking,* vol. 8(1), pp, 320-332, 2010.
[5] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," *Proc. ACM SenSys*, 2004.
[6] E. Meshkova, J. Ansari, D. Denkovski, J. Riihijarvi, J. Nasreddine, M. Pavloski, L. Gavrilovska, and P. Mahonen, "Experimental Spectrum Sensor Testbed for Constructing Indoor Radio Environmental Maps," *Proc. DySPAN*, 2011.
[7] X. Ning, "A Survey of Sensor Network Applications," in *IEEE Communications Magazine* 2002.
[8] J. L. Rodgers and W. A. Nicewander, "Thirteen Ways to Look at the Correlation Coefficient," *The American Statistician,* p. 18, 1988.
[9] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments," *Proc. ACM MobiCom*, 2001.
[10] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. M. Ni, "Side Channel: Bits over Interference," *Proc. ACM MOBICOM*, 2010.
[11] J. Yin, Q. Yang, and L. M. Ni, "Learning Adaptive Temporal Radio Maps for Signal-Strength-Based Location Estimation," *IEEE Transactions on Mobile Computing,* vol. 7 (7), p. 14, July 2008 2008.
[12] Z. Zhong, P. Chen, and T. He, "On-Demand Time Synchronization with Predictable Accuracy," *Proc. IEEE INFOCOM*, 2011.
[13] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "ZiFi: Wireless LAN Discovery via ZigBee Interference Signatures," *Proc. ACM MobiCom*, 2010.