Computer Communications xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Demodulation-free protocol identification in heterogeneous wireless networks ☆

Aijing Li [a], Chao Dong [a], Shaojie Tang [b], Fan Wu [c], Chang Tian [a,*], Bingyang Tao [c], Hai Wang [a]

[a] College of Communications Engineering, PLA University of Science and Technology, China
[b] Department of Information Systems, University of Texas at Dallas, TX, USA
[c] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China

ARTICLE INFO

Article history:
Received 3 July 2014
Received in revised form 20 August 2014
Accepted 24 August 2014
Available online xxxx

Keywords:
Protocol identification
Without demodulation
Heterogenous networks
USRP

ABSTRACT

Nowadays various wireless network protocols play respective roles to fulfill different demands. To better adapt to this heterogeneity and coexistence situation, it is critical for nodes to identify the available networks with high accuracy and low cost. Unlike traditional demodulation-based identification method, which is expensive and complexing, in this paper, we propose a novel conception called demodulation-free protocol identification. This method only employs the features of physical layer samples. We first extract features that can be used to identify different protocols. Specifically, a sparse sequence based Precision-Stable Folding Algorithm (PSFA) is proposed to detect periodicity feature, which is common in wireless network protocols. Then we construct a prototype with USRP to identify three commonly used protocols in the 2.4 GHz ISM band. Experiment results show that under low or moderate channel utilization, the accuracy is above 90%. We also show that the computational complexity is polynomial.

1. Introduction

The coexistence of heterogeneous networks has become a prominent trend, since various wireless network protocols play respective roles to fulfill different demands. In addition, most of the channels in these networks are overlapping with each other [1]. Take the city shown in Fig. 1 as an example. Wireless Sensor Networks (WSNs) are deployed in hospitals, forests, and roads for data collecting, e.g. CO_2, temperature, pollution, etc., while WiFi hotspots are deployed to provide Internet access in restaurants and campus. In addition, Wireless Personal Area Networks (WPANs) are used for short-distance communications, like smart home networks. In this context, to enhance coexistence and heterogeneity, it is essential for nodes to have a preliminary view of the wireless networks in current region. Therefore, *accurate and low-cost protocol identification* is playing an important role for quick media access and interoperability.

Traditional protocol identification schemes are demodulation-based. By demodulation and decoding received packets [2–5], the used protocols can be recognized. This requires nodes to implement all possible network protocol waveforms. The cost is high since physical layer (PHY) and most media access control (MAC) functions are implemented in hardware or firmware. Though Software Defined Radio (SDR) [6] can implement all possible waveforms in software and reduce the cost, nodes still need to load and try each waveform one by one [7]. Besides, packet decoding is not always feasible in practical circumstances, especially under war conditions. Various information technologies (e.g., information encryption) and electromagnetic interference (EMI) will be employed in future high-tech wars. In this situation, the SNR of received signals may drop to a level which cannot satisfy the demodulation requirement.

For above reasons, we are motivated to seek a less expensive protocol identification method, which can use PHY signals only and be demodulation-free. As we know, the current networks are based on a layered architecture, which results in the information scarcity of upper layer protocols when working with only PHY signals. Fortunately, protocol level behavior can be reflected to PHY signals, which leaves us a chance to infer upper-layer protocols through RF analysis. Its advantages are as follows:

- It can reduce the implementation cost. As only PHY signal features are used to recognize different protocols, there is no need to try each demodulation scheme, or implement the whole protocol stack of each potential protocol. This can greatly reduce the implementation complexity and financial cost.

☆ A preliminary version of this paper appeared in IEEE WCNC 2012, April 1–4, Paris, France.

* Corresponding author. Address: Box 110, 2 Biaoying, Yudao Street, Nanjing 210007, China.
E-mail address: tianchang126@126.com (C. Tian).

http://dx.doi.org/10.1016/j.comcom.2014.08.014
0140-3664/

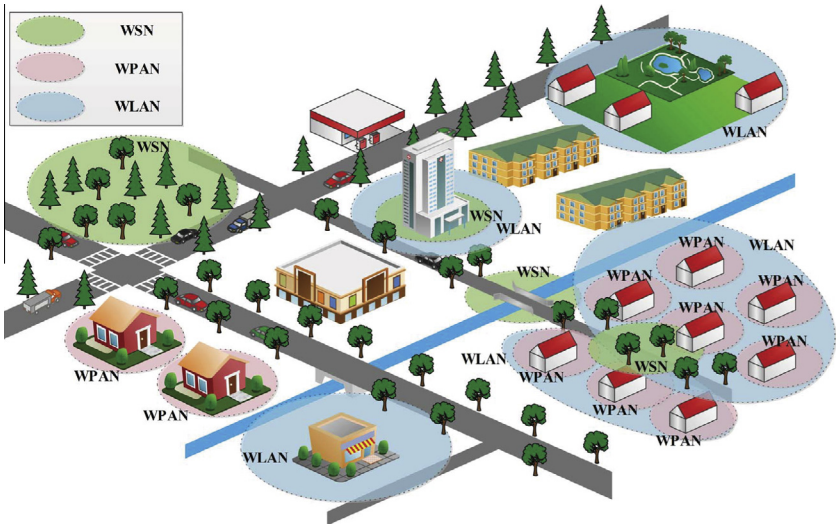

Fig. 1. Heterogeneous wireless networks in a city.

- It can reduce the computational cost. Compared with traditional identification approaches, some signal processing modules are not necessary, such as frequency offset compensation, phase offset compensation, and timing recovery. This reduces the computational complexity.
- It can be used in situations where reliable decoding is not feasible. For example, when scanning with omnidirectional antenna, the received SNR may be low for demodulation. We can first detect the existence of signals without demodulation. Then with beamforming and the direction of arrival estimation, received SNR can be strengthened and interested signals may be able to be demodulated.

Despite of the advantages, it may be more challengeable to consider raw PHY layer samples. Due to the layered architecture of networks, different layers work independently. Thus characterizing different signals and classifying them with these features can be difficult with original PHY layer samples.

Following the above idea, we propose a new conception called *demodulation-free protocol identification*, which only relies on PHY information. The key contributions of this paper can be summarized as follows:

- We propose the conception of demodulation-free protocol identification. It only employs features of PHY samples. This approach can be embedded into intelligent devices for network identification before media access, and provide interoperability across heterogeneous platforms.
- We investigate and extract the features of PHY signals that can be used to identify different wireless protocols. We analyze different signal features in both time domain and frequency domain. Specifically, a sparse sequence based Precision-Stable Folding Algorithm (PSFA) is proposed to detect the periodicity feature, which is common in wireless protocols [3,4].
- Taking three commonly used wireless protocols as an example, we construct a system design with USRP [8] to validate the feasibility and performance of the proposed conception. Experiments show that under low or moderate channel utilization ratio, the detection accuracy is above 90% for both single and multiple APs. We also show that the computational complexity is polynomial.

The remainder of this paper is organized as follows. Section 2 presents a review of related works. In Section 3, we investigate the features of different signals in both time domain and frequency domain. Section 4 describes the design and implementation of the identification system. The experimental results are shown in Section 5. Finally, Section 6 concludes the paper and presents future work.

2. Related works

Most of the protocol identification schemes are demodulation-based. By decoding and extracting information carried in the headers, we can obtain the necessary knowledge of the protocols used in each layer of the protocol stack. Protocol identification can be achieved either in an active or in a passive way. We introduce the two methods in the rest of this section.

2.1. Active protocol identification

Some of the existing systems solve the protocol identification problem by broadcasting active probing, for example, the beacon messages in most wireless protocols [3,4]. Kanuparthy et al. [9] investigate a user-level probing approach to detect and diagnose 802.11 pathologies. By introducing a probing server and probing client, detection and diagnosis can be done without any information from 802.11 devices and other link layer monitors. But this work is limited to only WiFi networks. Konark [10] is a service discovery and delivery protocol in Ad Hoc networks. Each device acts as a server and a client simultaneously. Clients use a discovery process known as active pull mechanism. Servers use an advertisement process to periodically announce their registered services. Then service can be discovered and delivered by pulling and advertising. Without a doubt, the broadcasting messages may introduce extra overhead to the network, which implies fewer transmitting opportunities for data packets and performance deterioration. Therefore, for the sake of performance, passive detection is preferable.

2.2. Passive protocol identification

The concept of Cognitive Gateway (CG) was proposed to promote interoperability across heterogeneous communication systems [11]. CGs can successfully classify four different types of wireless signals and provide corresponding communication services. The core design of CG is a Universal Classification Synchronization

Please cite this article in press as: A. Li et al., Demodulation-free protocol identification in heterogeneous wireless networks, Comput. Commun. (2014), http://dx.doi.org/10.1016/j.comcom.2014.08.014

(UCS) system. UCS can perform automatic signal recognition, synchronization, and provide necessary parameters for demodulation. Then with the identified demodulation mode as well as necessary information after demodulation and decoding, CGs can identify signals from different networks, and then detecting users' requests and routing their messages to the expected destination in heterogeneous communication systems.

WiBee [12] utilizes Zigbee sensors to build real-time WiFi radio maps, based on the observation that a Zigbee radio can sense WiFi frame transmissions although it cannot decode WiFi frames.

Miler et al. [13] proposed a method to identify Bluetooth devices and discover services with software defined radio platform. BlueID [14] is another practical system that identifies Bluetooth devices by fingerprinting their clocks. However, device-specific features may not be used in protocol identification directly.

DoF [15] and PinPoint [16] classify different signals by using the cyclic autocorrelation feature of the signals. However, this kind of method need antenna array to get the cyclic feature of the signals, which is expensive.

RFDump [17] is another signal-processing based approach which can identify the industrial, scientific and medical (ISM) band protocols for diagnostic purposes. RFDump use phase and timing analysis to identify the type of networks. Airshark [18] was designed to identify multiple non-WiFi RF devices running in the 2.4 GHz band with only off-the-shelf WiFi adapter, aiming to mitigate the interference to WiFi devices. SoNIC [19] is a system that enables sensor nodes to detect the type of interference they are exposed to. The key insight is that different interferers disrupt individual 802.15.4 packets in characteristic ways that can be detected by sensor nodes. Weng et al. [20] present a dimensionality reduction method to detect the interferences from microwave ovens, Wi-Fi and Bluetooth signals. However, these methods are technology-dependent.

In sum, most active and passive protocol identification methods are based on demodulation and decoding operation or analyzing the modulation based signal features, which is costly and inefficient. How to identify and analyze different protocols without demodulation have remained elusive. In this context, we propose the conception of demodulation-free protocol identification.

3. Feature extraction

To identify wireless network protocols without demodulation, we need to deeply unearth PHY signals of different protocols and extract the features that may be used to specify a certain protocol. This requirement entails two major questions:

(1) Among all the features of PHY signals, which of them should be exploited to reflect the upper-layer protocols?
(2) After analyzing and extracting the necessary features of different signals, how to use them to identify different protocols?

In this section, we analyze and characterize different PHY signals from various aspects in both frequency domain and time domain. After intensive research and theoretical analysis, we extract several features that can be used to identify a wireless network protocol, which are listed in Table 1.

Table 1
PHY signal features.

Frequency domain features	
Carrier frequency	Detecting protocols using orthogonal channels, like AM, FM radio and broadcast TV programme
Signal bandwidth	Estimating bandwidth of signals that share the same frequency resource
Frequency hopping	Detecting signals following FH mode, like Bluetooth
Time domain features	
Power distribution	Identifying spectrum etiquettes and communication patterns
Bit rate	Estimating transmitting bit rate
Cyclostationary feature	Detecting multicarrier signals, like OFDM
Time division	Detecting signals following TD mode, like 2G GSM systems

3.1. Carrier frequency and signal bandwidth

The most commonly used attributes to classify different signals in traditional static frequency assignment scheme are carrier frequency and signal bandwidth. Under static spectrum policies, spectrum resources are forced to behave like a fragmented disk. Different spectrum blocks are assigned to different protocols to avoid interference. Since different protocols work in orthogonal channels, frequency information is sufficient to specify a certain wireless service, such as AM, FM radio and broadcast TV programme. For the unlicensed 2.4 GHz ISM band, multiple wireless systems share the same frequency resource, meaning carrier frequency alone is not sufficient to classify them. However, different wireless standards define diverse bandwidth for communication. For example, channels used by WiFi span a bandwidth of 22 MHz [3], while Zigbee [4] and Bluetooth [5] use channels with 2 M and 1 M bandwidth respectively. Therefore, with signal location information in frequency domain we can coarsely classify different protocols.

3.2. Frequency hopping

Frequency Hopping (FH) technology is widely used in military radio communication equipment, as well as some civil communication systems, such as Bluetooth. Under FH scheme, signals hop from one channel to another according to predefined rules.

For example, Bluetooth is a typical frequency hopping protocol working in the 2.4 GHz ISM band. Bluetooth standard use Freuency-hopping Spread Spectrum (FHSS) transmission mode over 79 channels occupying a bandwidth of 1 MHz. The central frequencies are chosen according to the following equation:

$$F_c = 2402 + K \text{ MHz}, \quad K = 0, \ldots, 78$$

The hopping modes and hopping list can be helpful to recognize FH systems.

3.3. Power distribution

After making a close study of several communication standards, we found that the communication peers have to follow certain spectrum etiquettes and communication patterns. These patterns can be reflected in the received signal power distribution, which makes signal power distribution an important time domain feature for identifying different protocols. Take the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) signaling scheme defined in IEEE 802.11 as an example. It uses RTS (Request To Send) and CTS (Clear To Send) messages before data transmission to avoid collision (which is optional) and ACK messages to acknowledge correct receptions. SIFS (Short Interframe Space) is defined between data packets and ACKs in IEEE 802.11, as shown in Fig. 2. The signals are sent by commercial wireless adapter and received by USRP.

In addition, periodic broadcasting beacons are compulsory in many wireless networks in order to maintain global synchronization as well as to update state information. Periodic signals can

Please cite this article in press as: A. Li et al., Demodulation-free protocol identification in heterogeneous wireless networks, Comput. Commun. (2014), http://dx.doi.org/10.1016/j.comcom.2014.08.014

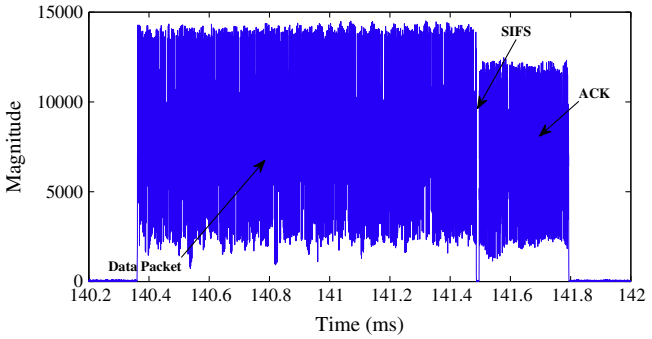

Fig. 2. The SIFS interval between data messages and ACKs can be obtained with a sample rate of 4 MHz.

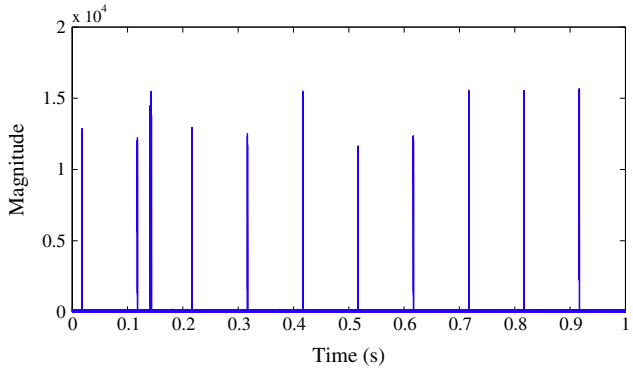

Fig. 3. Magnitude of 802.11 signals collected in 1 s with 4 kHz as its sample rate.

be found in received samples collected by a certain period of time, as shown in Fig. 3.

3.4. Bit rate

Different standards employ different transmitting bit rates, which can be considered as another feature of PHY signals. Bit rate difference results in the variation of packet interval and packet lasting time in the air. By observing the lasting time of packets with the same size (usually control messages like ACKs), we can distinguish different bit rate and further protocols.

3.5. Cyclostationary feature

Multicarrier technology, such as OFDM (Orthogonal Frequency Division Multiplexing), is adopted by some of the wireless standards. There are several ways to discriminate the single carrier and multicarrier signals in the literature [21,22]. Besides, OFDM signal itself has some unique features. Under OFDM scheme, transmitted data is modulated to several different subcarriers and transmitted simultaneously. CP (Cyclic Prefix) is used to overcome ICI (Inter Carrier Interference) by duplicating the end of the OFDM symbol and adding the repetition to the front. CP can be used for synchronization and blind signal identification. Let $y(n)$ be the received sample sequence. From [21] we know the time vary correlation of $y(n)$ is

$$r_y(n,\tau) = y(n)\cdot y^*(n-\tau) = \sum_{m=0}^{L-1}\sigma^2_{h(m)}e^{j2\pi\Delta f\tau}\sigma_s^2\sum_{k=0}^{N-1}e^{j\frac{2\pi}{N}k\tau}\cdot\sum_{l=-\infty}^{\infty}[g'[n - lP]g'^*[n-lP-\tau]] + r_w(\tau) = r_y(n+kP,\tau) \quad (1)$$

where $\sigma^2_{h(m)}$ is the variance of channel impulse response, σ_s^2 is the variance of information symbols, $g[n]$ is the transmitter pulse shaping filter, Δf is the frequency offset, $w(n)$ is the white Gaussian noise and P is defined as $P = N + L$, where N is the number of subcarriers and L is the channel order.

The autocorrelation function of the receive signal is defined as [22]

$$E\{y(n)y^*(n+\tau)\} = \begin{cases} \sigma_s^2+\sigma_w^2 & \tau = 0 \\ \sigma_s^2 e^{-j2\pi\epsilon} & \tau = N_u \\ 0 & \text{other} \end{cases} \quad (2)$$

By autocorrelation with OFDM symbols, we can find three peaks from the correlation results, locating at $\tau = 0,\ \pm N_u$. The maximum value is achieved when $\tau = 0$. Here N_u represents the length of useful symbols.

3.6. Time division

Some of the wireless systems access the wireless media in a time division manner, for example, Bluetooth standard and GSM system. Bluetooth uses a mixture of Time Division Duplex (TDD) and FHSS. Its transmission time is divided into time slots with a length equal to 625 μs. The transmission can use up to 5 time slots. GSM system is another commonly used TDMA system. Each frame is divided into eight radio timeslots and the frame duration is 4.615 ms. For signals conforming to time division duplexing, the start time of each transmission always shows a pattern

$$S(i+1) - S(i) = kT_s, \quad k = 1, 2, \ldots$$

where $S(i)$ is the start time of the ith transmission, and T_s is the duration of a time slot.

3.7. Summary

In this section, we mainly analyze and extract features that may be obvious or hidden in signals. Extracting these features is an important factor for the success of wireless protocol identification. Some of these features are specific to a certain protocol, like carrier frequency and bandwidth under static spectrum assignment policy, while some of them need to be combined with other features to identify a protocol. We can use different combinations of these features to detect different protocols. With the proliferation of wireless communication technologies, more features will be extracted and exploited. We leave this to future work.

4. System design and implementation

With the extracted signal features in both frequency domain and time domain mentioned above, it is rational to identify and analyze different protocols in physical layer without demodulation operation. To validate the feasibility of our demodulation-free identification method, we design a prototype to identify three commonly used protocols in the 2.4 GHz ISM band as an example, including WiFi, Zigbee and Bluetooth. It is worth noting that our conception is not limited to these three protocols.

4.1. Overview

After investigating those three standards, we extract and combine the features that can be used to identify them. Table 2 shows those features of each wireless standards. In the table, '✓' means it is a candidate feature to identify this standard, while '✗' means the opposite.

In the rest of this section, we mainly discuss how to exploit these features to achieve our goal. Notice that most of the features are not unique to a specific protocol, so we need to consider the combination of these features.

Please cite this article in press as: A. Li et al., Demodulation-free protocol identification in heterogeneous wireless networks, Comput. Commun. (2014), http://dx.doi.org/10.1016/j.comcom.2014.08.014

Fig. 4 illustrates the architecture of our design. The system block diagram includes all the modules implemented in this prototype. We will describe each module in more details in the remaining of this section. The entire structure of our prototype consists of five blocks, including preprocessing, bandwidth estimation, time division detection, periodic detection and multicarrier detection, each of which employs one of the aforementioned features we have extracted.

4.2. Platform

We choose the well accepted SDR platform USRP and open source software toolkit GNU Radio to implement our system. In our experiment, we choose the RFX2400 daughterboard, which can cover the spectrum band with a range of 2.3–2.8 GHz. We use a laptop and a PC equipped with Atheros AR5001X+ wireless network adapter to provide a WiFi service, TelosB motes equipped with CC2420 radio to provide a Zigbee service and BT520 Bluetooth adapters to provide a Bluetooth service.

4.3. Preprocessing

Algorithm 1. Denoising

Input:
R – received samples
α – signal power threshold
w – window size
Output:
R_{out} – received signals with noise elimination
R_{pre} – preprocessed signals
S – the start position of each signal
E – the end position of each signal
Process:
1: **for** $i = 1 : w : N - w$ **do**
2: **if** $\frac{1}{w}\sum_{k=i}^{i+w} R^2[k] > \alpha$ **then**
3: $R_{pre}[i : i + w] = 1$
4: **else**
5: $R_{pre}[i : i + w] = 0$
6: $R_{out}[i : i + w] = 0$
7: **end if**
8: **end for**
9: $S = \{i | R_{pre}[i - 1] = 0 \wedge R_{pre}[i] = 1\}$
10: $E = \{i | R_{pre}[i - 1] = 1 \wedge R_{pre}[i] = 0\}$

Preprocessing block is needed to eliminate noise and interference from received signals before further processing. Signals can be easily separated from noise by energy detection, since the signals are with higher received energy, as is shown in Fig. 2. The main idea of energy detection is to compute the average energy of the samples within a window. Only if the average energy is greater than a predefined threshold, we decide there is a signal present.

Table 2
Features for each standard.

	WiFi		Bluetooth	Zigbee
	802.11b	802.11g/n	802.15.1	802.15.4
Carrier frequency	✓	✓	✓	✓
Signal bandwidth	✓	✓	✓	✓
Time division	✗	✗	✓	✗
Periodic	✓	✓	✗	✗
Multicarrier	✗	✓	✗	✗

$$P = \sum_{n=1}^{N} (Y(n))^2 \begin{cases} > Threshold, & signal\ present \\ < Threshold, & signal\ absent \end{cases} \quad (3)$$

Besides noise elimination, preprocessing block also calculates and maintains necessary parameters which are crucial for the subsequent blocks. The main process of this block is shown in Algorithm 1.

Noise elimination and preprocessing are first accomplished by energy detection (lines 1–8). Then the starting and ending point of each transmission is derived by finding the sudden jump of the signal energy (lines 9–10). The threshold α is important to eliminate the noise. To improve the efficiency of threshold based energy detection algorithm, different methods to determine the threshold are proposed, such as dynamic threshold [23] and optimal threshold determination [24]. For simplicity, we set the detection threshold α to be $2 * P_N$, where P_N is the power level of noise. The outputs are delivered to the subsequent blocks for obtaining other parameters and protocol identification. S, E and R_{out} are used to estimate the bandwidth of each burst, and R_{pre} helps the periodic detection.

4.4. Bandwidth estimation

Bandwidth estimation plays an important role in the process of protocol identification. Since wireless standards usually employ predefined bandwidths, it is possible to make bandwidth a criterion to coarsely classify different standards. We implement the bandwidth estimation block to estimate the bandwidth of detected signals. Fig. 5 shows the spectrogram of a snapshot centered at 2.41 GHz. Due to the limitation of USRP1, it is impossible to obtain signals with a bandwidth of more than 8 MHz, so only 4 MHz signals are shown in Fig. 5. We can immediately find that there is a periodic signal spanning at least 4 MHz in bandwidth, which is considered to be the wideband WiFi signal. There are still some other narrowband bursts through the spectrogram. These bursts spanning about 2 MHz in bandwidth are in fact Zigbee signals transmitted by sensors.

Some methods for bandwidth estimation are proposed in the literature. For simplicity, we use the measurement method of occupied bandwidth ($\beta\%$ method) [25]. The power below the lower and above the upper frequency limits are each equal to a specified percentage $\beta/2$ of the total mean power. The value of $\beta/2$ should be taken as 0.5%.

After bandwidth estimation, the input signals can be coarsely classified to three categories according to their bandwidth. Then these three kinds of signals are conveyed to different branches respectively for further analysis.

4.5. TD detection

Signals with a bandwidth of 1 MHz are considered to be transmitted by a Bluetooth device and conveyed to the Bluetooth detection branch. TD detection is used to detect whether the signals exhibit a time division pattern.

Under the time division duplexing standard, each transmission starts at the beginning of a slot. We maintain the starting time $S(i)$ and ending time $E(i)$ of the latest several bursts to detect the time division pattern. We believe the signals follow a time division protocol, if the interval of two consecutive bursts matches the pattern

$$\begin{cases} S(i+1) - S(i) = kT_s \\ E(i) - S(i) = mT_s, \quad k, m = 1, 2, \ldots \end{cases} \quad (4)$$

Here, T_s is the length of a time slot. Specifically for bluetooth systems, T_s is 625 μs and m is an integer within $[1, 5]$.

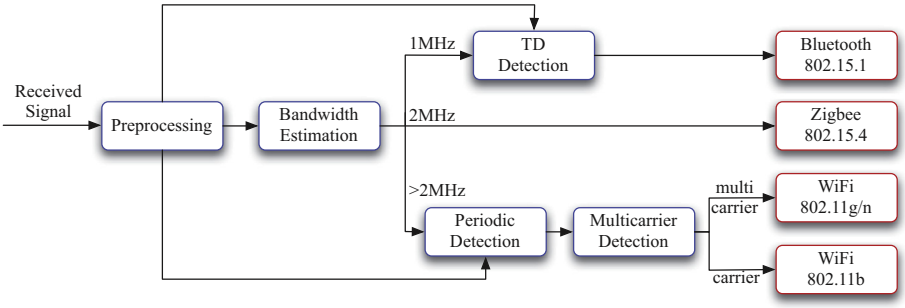

Fig. 4. System block diagram.

4.6. Periodic detection

Periodic beacons are indispensable in WiFi networks as they are of great significance to update station information and maintain synchronization. This makes periodicity an important feature for WiFi identification. Without demodulation, we cannot be aware of the content of the beacons, however, we still can recognize their existence. We detect the presence of periodic beacons with PHY samples in time domain, where FFT is not necessary. In order to discover the periodic signals embedded in the received sample sequence, we adopt the folding algorithm, which was first proposed for the detection of periodic pulse trains and pulsar search.

The main idea of folding can be explained in Fig. 6. Consider a received sample sequence containing N elements $R[n]$ with a signal of period P embedded in it. The correlation could be performed by additively folding the data on itself with period P. The result would consist of P elements $F[i]$, where

$$F[i] = \sum_{j=0}^{\lfloor N/P \rfloor - 1} R[i + j \cdot P], \quad i = 1, \ldots, P \tag{5}$$

After folding at the correct period, periodic signals can be separated from noise and other signals. The reason lies in the fact that folding can amplify the periodic signals and obtain a peak value, while the noise and interference get lower value due to the lack of periodicity.

Since the period of the beacons is unknown, we need to search all the possibilities, which can be costly. Ref. [26] proposed a novel algorithm to reduce the on-line searching consumption. However, the cost of off-line CMF (Common Multiple Folding) tree conducting time and computing complexity is also unbearable. Besides, the basic folding algorithm and CMF tree can only deal with integer period. However, a decimal period is possible because of different sample rate. For example, with a sample rate of 4 K/s, a period of 100 ms corresponds to 409.6 samples. To cope with decimal period and reduce the additions in the folding, Fast Folding Algorithm (FFA) [27] was proposed. FFA can search for non-integer period between P_0 and $P_0 + 1$. But the precision of FFA is not stable as the step size is defined to be P_0/N. So the precision is limited by the starting period P_0. Besides, the sequence length N and P_0 have to satisfy relation $log_2 N/P_0 \in N^+$, which leads to the insufficient use of received samples. To overcome these shortcomings, we propose a sparse-sequence based Precision-Stable Folding Algorithm (PSFA). With PSFA, we can detect decimal periodic with any specified precision.

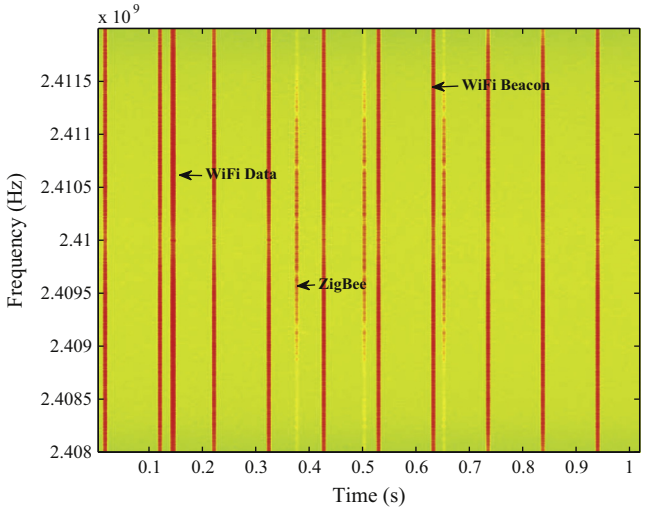

Fig. 5. Spectromgram of the ISM band centered at 2.41 G and spanning 4 MHz.

The operation of PSFA is shown in Fig. 7. From the operation, we can find many samples like 'A' and 'B' which span two elements in the folding results. We set different weights to mark the contributions of each sample to the folding result. For each sample $R[i]$, the offset is defined as $\Delta_i = i - \lfloor i/P \rfloor \cdot P$, and we set.

$$u_i = \lfloor \Delta_i \rfloor$$

$$r_i = \Delta_i - u_i$$

The contribution of the ith sample to the kth element of folding result is

$$C_{i,k} = \begin{cases} 1 - r_i, & k = u_i \\ r_i, & k = (u_i + 1) mod \lceil P \rceil \\ 0, & \text{others} \end{cases} \tag{6}$$

Then the weighted contribution can be expressed as

$$Contrib_{i,k} = R[i] \cdot C_{i,k} \tag{7}$$

The folding result of PSFA consists of $\lceil P \rceil$ elements $\mathcal{F}[j]$, where

$$\mathcal{F}[j] = \sum_{i=1}^{N} Contrib_{i,k} \tag{8}$$

From Eq. 7, we notice when $R[i] = 0, \forall k, Contrib_{i,k} = 0$. After the eliminating noise and data from the received samples, There are only beacons in the sequence, so the received sequence can be

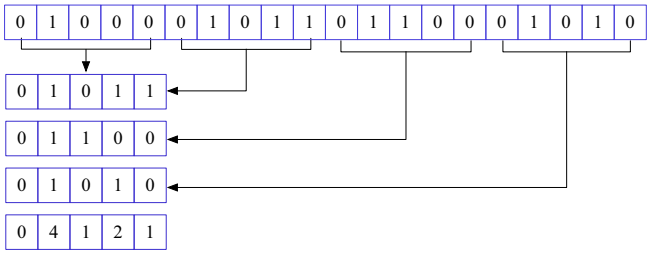

Fig. 6. Operation of Folding Algorithm. In this time series, there is a periodic symbol with period 5 at position 2. After FFA, there is a peak with magnitude 4 at position 2.

Please cite this article in press as: A. Li et al., Demodulation-free protocol identification in heterogeneous wireless networks, Comput. Commun. (2014), http://dx.doi.org/10.1016/j.comcom.2014.08.014

converted to a sparse 0–1 sequence. With sparse sequence, we can reduce the number of additions from $N - P$ to γN, where

$$\gamma = \frac{|\{R[i]|R[i] = 1\}|}{|R|}$$

is defined as channel utilization ratio. The noise signal can be eliminated in the preprocessing block and data packets can be removed by erasing all the bursts whose lasting time is out of the range of a beacon frame's in-air time.

Algorithm 2. Periodicity Detection

Input:
R_{pre} – samples after preprocessing
P – set of possible beacon periods
α – threshold
Output:
P_{out} – possible period of beacon
Process:
1: eliminate data from R_{pre}
2: **while** True **do**
3: Compute the folding result $F_{p[i]}$ of each $P[i]$
4: // find the max normalized folding result of each $P[i]$
5: $F\_max = P[i] \cdot max\{F_{p[i]}\}$
6: sort F_max in the descending order, as well as corresponding P and F_p
7: **if** $F\_max[0] \geqslant \alpha$ **then**
8: $P_{out}[k{+}{+}] = F\_max[0]$
9: remove detected beacons from R_{pre}
10: **else**
11: break
12: **end if**
13: **end while**

By setting different searching step size, we can detect signals period with specific precision. The pseudo code of periodicity detection is shown in Algorithm 2. Before PSFA process, the data packets need to be eliminated from the preprocessed signals to obtain a sparse sequence (line 1), since data packets may cause false peaks after folding. Then we calculate the folding result of each possible period according to formula (5) (line 3). The folding peak of each period is normalized by multiplying the period $P[i]$ (lines 5–6). P can be found by comparing the maximum folding peak to a certain threshold (lines 7–9). After recognizing one possible period, we eliminate the beacon signals from R and move to next round until no other periodic signals are found (lines 10–13).

In practical scenarios, the typical beacon interval used in most wireless cards is between 100 and 200 ms and is set to 100 ms by default (like the Atheros AR5001X+, TP-link TL-WN322G+ and most wireless network adapters used in Acer laptops). So here we only considered the range of [80,120] ms as possible periods. The folding result of the received WiFi signals is shown in Fig. 8. Fig. 8(a) shows the magnitude of received samples collected in 1 s. We notice that the periodic beacons are evident. Fig. 8(b) and (c) shows the folding result under the correct beacon interval (100 ms in our experiment) and the wrong beacon interval (110 ms in our experiment) respectively. After folding at the right beacon period, we can get a peak value equal to 10, while folding at the wrong period only gets some loose points with magnitude of 1 or 2.

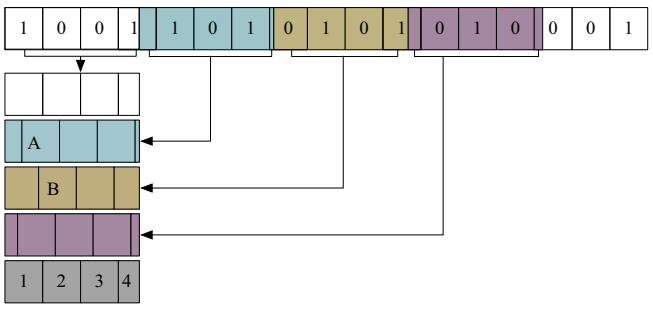

Fig. 7. Operation of Precision-Stable Folding Algorithm. The sequence is folded at a decimal period of 3.6.

4.7. Multicarrier detection

In order to further identify the detected wideband signals, namely the 802.11g/n signals in Fig. 4, we make use of the cyclostationary feature of multicarrier signals. As mentioned above, OFDM signals exhibit clear 3-peak feature by autocorrelation operation. Since it is impossible to obtain 20 MHz WiFi signals with USRP1 due to the speed of USB 2.0 interface, we use narrowband OFDM signal to verify the feasibility of multicarrier detection. We consecutively send 2 MHz OFDM signals with USRP and then analyze the received samples, while make other OFDM parameters exactly the same as that are defined in the IEEE 802.11 g standard. Fig. 9 shows the normalized autocorrelation result of received OFDM signals. From Fig. 9, we can see one peak in the center and two lower peaks symmetrically distributed on both sides of the highest one. The three peaks validates the feasibility of OFDM signal detection.

5. Validation and discussion

Due to the speed of USB 2.0 interface, USRP1 can support a sample rate of no more than 8 MHz. So the prototype implements all the blocks in Fig. 4 except for the multi-carrier detection block. We let the wireless adapter, sensors and bluetooth adapter work simultaneously. The signals are collected by USRP and then conveyed to the identifying blocks. Energy detection block detects the presence of signals. Then the detected signals are classified to three categories by bandwidth estimation block for further processing. Signals with bandwidth of 1 MHz are delivered to TD detection block for bluetooth identification, while those wideband signals are conveyed to periodic detection block for WiFi identification. Experimental results show that it can successfully identify WiFi, Zigbee and Bluetooth signals.

5.1. Accuracy

From the system block diagram in Fig. 4, we can see bandwidth estimation plays an important role in the identification process. The identification accuracy of Bluetooth and Zigbee mainly relies on the accuracy of bandwidth estimation algorithm. A high-performance bandwidth estimation algorithm can improve the accuracy of identification. Besides, signal bandwidth estimation is noise independent. As long as Bluetooth and ZigBee bursts are not totally overlapped by WiFi signals, bandwidth estimation block can recognize them. Thus, in this subsection, we mainly discuss the accuracy of WiFi identification. We evaluate the accuracy of WiFi identification under two different circumstances, both with single and multiple APs.

5.1.1. Single AP detection

From Fig. 8(a) we can see there are other data transmissions except for periodic beacon signals. If there exists a system with heavy data traffic, it may cause other peaks after folding, leading to false detection result. To evaluate the detection accuracy with

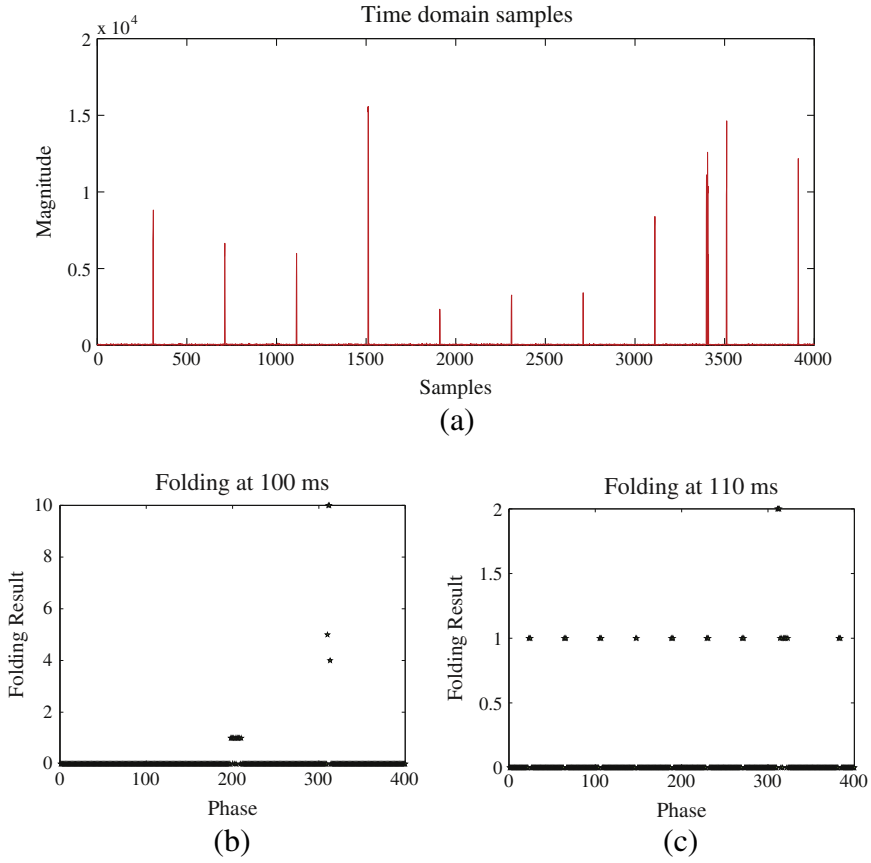

Fig. 8. (a) The magnitude of collected samples in 1 s. (b) The folding result at the correct period. (c) The folding result at the wrong period.

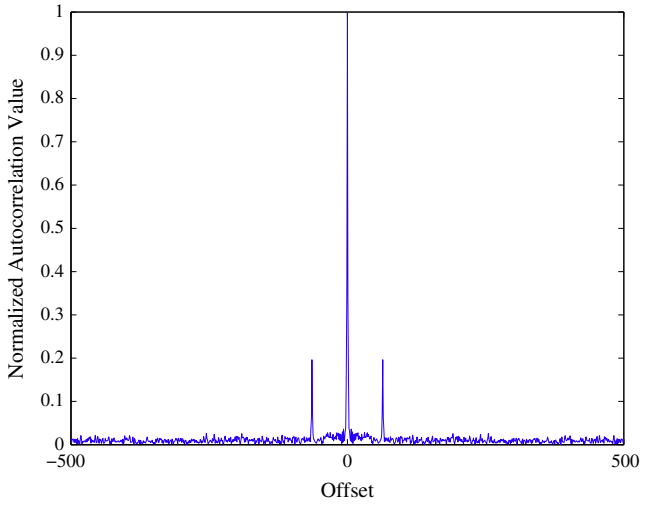

Fig. 9. Autocorrelation result of received OFDM signals.

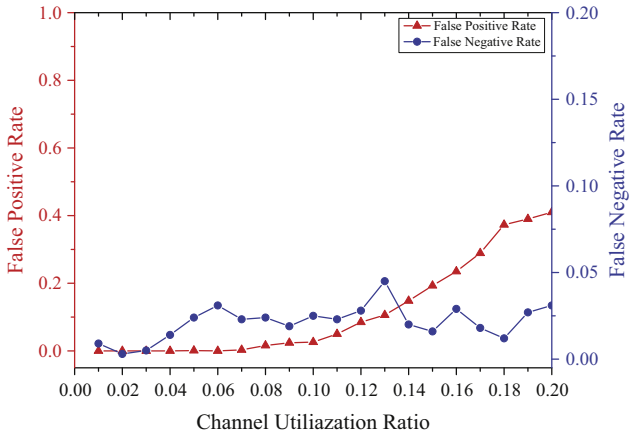

Fig. 10. Detection accuracy vs. channel utilization.

data interference, we start a FTP service between the AP and the client to generate a data traffic. Fig. 10 shows the detection accuracy of single AP under different channel utilization rate. Both False Negative (FN) and False Positive (FP) rate keep low when the channel utilization is under 0.1. The FN rate keeps in a low level, while the FP rate raise sharply with the increase of channel utilization. This is reasonable since data transmission may cause false folding peaks. However, in practical WiFi systems the channel utilization ratio is usually low. According to traces collected at SIGCOMM 2008 [28], the average ratio is 7.6%, under which circumstance the FP and FN rate are 0.3% and 2.3% respectively.

5.1.2. Multiple APs detection

In practical wireless environment, multiple APs with different signal power and beacon period may coexist with a high probability. By iteratively eliminate the detected signals, We expect to recognize them all (see Algorithm 2). We carry out this experiment inside a building. We placed three D-Link routers at different places in the lab. The experimental parameter is set according to Table 3. AP1 is set with fixed beacon interval, which is 100 ms. The beacon intervals of AP2 and AP3 are changeable. By moving

Please cite this article in press as: A. Li et al., Demodulation-free protocol identification in heterogeneous wireless networks, Comput. Commun. (2014), http://dx.doi.org/10.1016/j.comcom.2014.08.014

Table 3
Experimental parameter.

	Group 1			Group 2			Group 3
	AP1	AP2	AP3	AP1	AP2	AP3	AP1
Signal power (dBm)	−72	−39	−42	−52	−31	−37	−54
Beacon period (ms)	100	[80,120]	[80,120]	100	[80,120]	[80,120]	100

Table 4
Detection accuracy in multiple APs environment.

	Experiment time (ms)	False positive (%)	False negative (%)	Detected accuracy (%)
Group 1	120,000	3.57	6.02	90.4
Group 2	120,000	2.11	4.08	93.8
Group 3	120,000	0.00	0.00	100.

the receiving USRP, we can obtain different received signal power. APs in Group 1 have larger signal power variance. Single AP detection is also carried out in Group 3 as a reference.

Table 4 illustrates the experiment results. We can conclude from the results that multiple APs may cause interference with each other, and larger power difference may raise false positive and false negative. However, the detection accuracy still reaches 90.4%.

5.1.3. Detection accuracy under low SNR

In practical environment, the SNR of received signals might be low, so that demodulation is not feasible. To validate the applicability and effectiveness of demodulation-free method in the scenario where SNR is insufficient for demodulation, we conduct the experiments to compare the accuracy of demodulation-dependent and demodulation-free method. Fig. 11 shows the experiment result. From Fig. 11, we can see that demodulation-free detection can achieve high accuracy even when the SNR is low for demodulation. The demodulation-free method can achieve over 90% accuracy when SNR is 8 dB while it needs over 11 dB for demodulation. The SNR requirement for demodulation-free method is much lower than that of demodulation-dependent method.

5.2. Complexity

In this section, we firstly analyze the complexity of proposed PSFA. As mentioned in Section IV, PSPA can reduce the number of additions to γN for a single period detection, where γ is the channel utilization ratio. Then the additions required for searching a period set P with precision δ is $\frac{|P|}{\delta} \cdot \gamma \cdot N$. Hence the computational complexity of PSFA is polynomial.

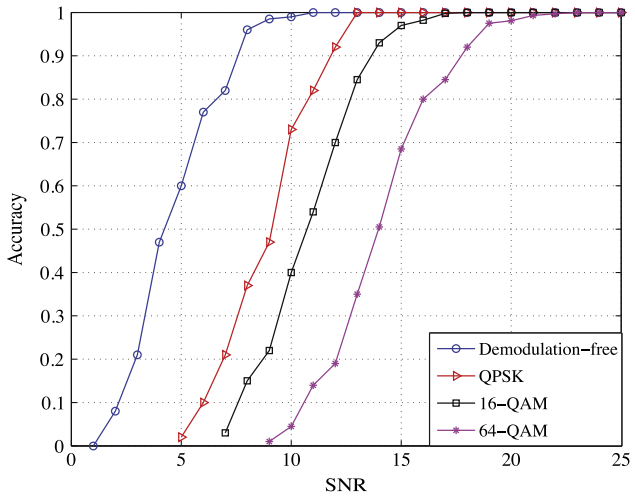

Fig. 11. Detection accuracy between demodulation-free and demodulation-dependent method under low SNR.

Table 5
Computational cost of different folding algorithms.

	Basic folding	FFA	PSFA
Complexity	$\sum_{P_i \in P}(N - P_i)$	$\sum_{P_i \in P} N' \cdot \log_2(N'/P_i)$	$\gamma N \cdot \frac{\lvert P \rvert}{\delta}$
$P_0 = 80$ $m = 80$ $\gamma = 0.015$	1.5×10^5	4.2×10^5	2.4×10^4
$P_0 = 80$ $m = 80$ $\gamma = 0.2$	1.5×10^5	4.2×10^5	3.2×10^5
$P_0 = 80$ $m = 160$ $\gamma = 0.015$	3×10^5	8×10^5	4.8×10^4

Table 5 illustrates the comparison result of our proposed PSFA and other folding algorithms. We calculate the complexity with $N = 2000$ samples, different channel utilization ratio γ, start period P_0 and period range m. The precision δ of PSFA is set to 0.1.

From Table 5 we notice that the computational cost of basic folding algorithm is always lower than FFA. That is because the step size of basic folding algorithm is lager, say 1, since it can only deal with integer period. The cost of PSFA is much less than FFA, and for low channel utilization, the cost is less than basic folding algorithm. As mentioned above, the practical channel utilization is relatively low and with preprocessing and denoising operation, the value of γ can be even lower. Thus, our proposed PSFA is superior to other choices.

In our prototype, the time complexity of preprocessing is $O(N)$, where N is the number of samples. The complexity of bandwidth estimation is equal to the complexity of FFT, which is $O(N \cdot \log N)$. The complexity of periodic detection is mainly determined by the PSFA algorithm, which would be $\frac{|P|}{\delta} \cdot \gamma \cdot N$, where P is the set of possible period set and δ is the stepsize or precision of the detected period. Thus, protocol identification can be accomplished within polynomial time.

Compared with the demodulation-based identification, the complexity and implementation cost of our approach is much lower. Under demodulation-based schemes, more modules are required before the demodulation operation, such as frequency offset estimation, phase offset compensation, timing recovery, and preamble synchronization. Thus, the cost of our approach is less than the demodulation-based schemes.

6. Conclusions and future work

In this paper, we present a new conception called demodulation-free protocol identification method. This concept can provide useful message for intelligent devices to make media access decisions and enhance interoperability across heterogeneous platforms. We implement a prototype with USRP which successfully identified three most commonly used wireless standards in

Please cite this article in press as: A. Li et al., Demodulation-free protocol identification in heterogeneous wireless networks, Comput. Commun. (2014), http://dx.doi.org/10.1016/j.comcom.2014.08.014

2.4 GHz ISM band to validate the feasibility of proposed conception.

However, the application of this conception is not limited to the above mentioned protocols. It can be easily extended to identify and analyze other wireless network protocols with these PHY features. Besides, with the proliferation of new techniques and wireless protocols, more features that can be used to specify a protocol need to be extracted from PHY signals to cope with more complex situations. Further, these features can provide useful information for device to make better media access decision in cognitive networks. With the deployment of cognitive radio networks, more user-defined protocols may be used. These protocols may cause interference to existing networks and need to be identified. We leave these as future work.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61103224 and 61201217) and the Natural Science Foundation of Jiangsu Province (BK2011118).

References

[1] F.C. Commission, Title 47-telecommunication, chapter 1, part15-radio frequency devices, Tech. rep., U.S. Goverment Printing Office, October 2010.
[2] Wireshark, 2014 <http://www.wireshark.org/>.
[3] Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Tech. rep., IEEE Std. 802.11, 2012.
[4] Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs), Tech. rep., IEEE Std. 802.15.4, 2006.
[5] Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs), Tech. rep., IEEE Std. 802.15.1, 2005.
[6] E. Grayver, Implementing Software Defined Radio, Springer, New York, 2013.
[7] B. Le, T.W. Rondeau, D. Maldonado, C.W. Bostian, Modulation identification using neural network for cognitive radios, in: Software Defined Radio Forum Technical Conference, 2005.
[8] Ettus Research, 2014 <http://www.ettus.com/>.
[9] P. Kanuparthy, C. Dovrolis, K. Papagiannaki, S. Seshan, P. Steenkiste, Can user-level probing detect and diagnose common home-WLAN pathologies, in: ACM SIGCOMM Computer Communication Review, vol. 42, 2012, pp. 7–15.
[10] S. Helal, N. Desai, V. Verma, C. Lee, Konark-a service discovery and delivery protocol for ad-hoc networks, in: Wireless Communications and Networking (WCNC), vol. 3, 2003, pp. 2107–2113.
[11] Q. Chen, Cognitive Gateway to Promote Interoperability, Coverage and Throughput in Heterogeneous Communication Systems, Ph.D. thesis, Virginia Polytechnic Institute and State University, 2009.
[12] W. Li, Y. Zhu, T. He, WiBee: Building WiFi radio map with ZigBee sensor networks, in: Proceedings IEEE INFOCOM, 2012, pp. 2926–2930.
[13] R. Miller, W. Xu, P. Kamat, W. Trappe, Service discovery and device identification in cognitive radio networks, in: IEEE Workshop on Networking Technologies for Software Define Radio Networks, 2007, pp. 40–47.
[14] J. Huang, W. Albazrqaoe, G. Xing, BlueID: a practical system for bluetooth device identification, in: Proceedings IEEE INFOCOM, 2014.
[15] S. Hong, S. Katti, DOF: a local wireless information plane, in: ACM SIGCOMM Computer Communication Review, 2011.
[16] K. Joshi, S. Hong, S. Katti, PinPoint: localizing interfering radios, in: Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, 2013.
[17] K. Lakshminarayanan, S. Sapra, S. Seshan, P. Steenkiste, RFDump: an architecture for monitoring the wireless ether, in: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, 2009, pp. 253–264.
[18] S. Rayanchu, A. Patro, S. Banerjee, Airshark: detecting non-WiFi RF devices using commodity WiFi hardware, in: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, 2011, pp. 137–154.
[19] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Nordén, P. Gunningberg, SoNIC: Classifying interference in 802.15.4 sensor networks, in: Proceedings of the 12th International Conference on Information Processing in Sensor Networks, 2013, pp. 55–66.
[20] Z. Weng, P. Orlik, K.J. Kim, Classification of wireless interference on 2.4 GHz spectrum, in: IEEE WCNC, 2014.
[21] B. Park, H. Cheon, E. Ko, C. Kang, D. Hong, A blind OFDM synchronization algorithm based on cyclic correlation, in: IEEE Signal Processing Letters, vol. 11, 2004, pp. 83–85.
[22] P. Liu, B. Li, Z. Lu, F. Gong, A blind time-parameters estimation scheme for OFDM in multi-path channel, in: International Conference on Wireless Communications, Networking and Mobile Computing, vol. 1, 2005, pp. 242–247.
[23] G. Yu, C. Long, M. Xiang, W. Xi, A novel energy detection scheme based on dynamic threshold in cognitive radio systems, in: Journal of Computational Information Systems, vol. 8, 2012, pp. 2245–2252.
[24] X. Zhai, H. He, G. Zheng, Optimization of threshold for local spectrum sensing with energy detector, in: Journal of Shanghai University (English Edition), vol. 15, 2011, pp. 132–136.
[25] Bandwidth measurement at monitoring stations, Recommendation, ITU-R SM.443-4, 2007.
[26] Y. Xiong, R. Zhou, M. Li, G. Xing, L. Sun, J. Ma, ZiFi: Exploiting cross-technology interference signatures for wireless LAN discovery, in: IEEE Transactions on Mobile Computing, vol. 99, 2014.
[27] D. Staelin, Fast folding algorithm for detection of periodic pulse trains, Proceedings of the IEEE, vol. 57, IEEE, 1969, pp. 724–725.
[28] Sigcomm 2008 Traces <http://www.cs.umd.edu/projects/wifidelity/sigcomm08traces/>.