Wireless Networks

Gaotao Shi
Keqiu Li

Signal Interference in WiFi and ZigBee Networks

Springer

Wireless Networks

Series editor

Xuemin (Sherman) Shen
University of Waterloo, Waterloo, Ontario, Canada

More information about this series at http://www.springer.com/series/14180

Gaotao Shi · Keqiu Li

Signal Interference in WiFi and ZigBee Networks

Springer

Gaotao Shi
School of Computer Science
and Technology
Tianjin University
Tianjin
China

Keqiu Li
School of Computer Science
and Technology
Dalian University of Technology
Dalian
China

ISSN 2366-1186 ISSN 2366-1445 (electronic)
Wireless Networks
ISBN 978-3-319-47805-0 ISBN 978-3-319-47806-7 (eBook)
DOI 10.1007/978-3-319-47806-7

Library of Congress Control Number: 2016954523

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Chapter 1
Introduction

Signal interference is the number one enemy of wireless systems. Recently, as wireless systems proliferate worldwide, mutual interference between heterogeneous wireless networks has drawn much attention of researchers. The commonly used short communication technologies in Internet of Things (IoT) WiFi and ZigBee share the same frequency spectrum, and must coexist in extremely complicated signal environments.

1.1 Signal Interference

Wireless transmission depends on a radio frequency (RF) carrier signal whose frequency varies with different communication technologies, but all have been assigned by some institution such as the Federal Communications Commission (FCC) in USA.

The data signals will be modulated onto the RF carrier signal before emitted into air in the form of electromagnetic waves. When multiple different modulated signals arrive at the receiver, the useful signal for the receiver will be distorted or disrupted. If the received signals have the same or close frequency, interference is unfortunately very possible. The interference will decrease the received signal strength. A worst case of interference is illustrated in Fig. 1.1 where two signals of the same frequency are out of phase with each other, resulting in a net reduction in the overall level of the combined signal. If two identical signals are 100 % or 180 degrees out of phase they will completely cancel one another if combined. Thus signal interference will happen within the communication system using the same communication technology as well as those using different technologies as long as they are with close frequency.

G. Shi and K. Li, *Signal Interference in WiFi and ZigBee Networks*,
Wireless Networks, DOI 10.1007/978-3-319-47806-7_1

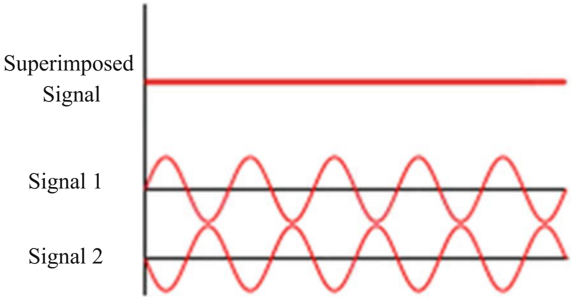


Fig. 1.1 Illustration of signal interference with the same frequency

1.2 WiFi and ZigBee

As the most popular way for accessing the internet, WiFi is almost ubiquitous in office buildings, homes, and even outdoors in urban areas and embedded in many kinds of portable digital devices. In WiFi networks, the IEEE 802.11 protocol family defines the PHY layer and MAC layer for implementing wireless local area network (WLAN) computer communication in the 2.4 and 5 GHz frequency bands. Figure 1.2 illustrates the channels for IEEE 802.11 in 2.4 GHz frequency band. Since only WiFi working in 2.4 GHz has the possibility to interfer ZigBee networks, we don’t consider that in 5 GHz.

ZigBee is a very attractive technology for implementing low-cost, low-power wireless control networks requiring high flexibility in node placement. The technology is defined by ZigBee specification maintained and published by ZigBee Alliance. ZigBee is an IEEE 802.15.4-based specification which defines the ZigBee PHY layer and MAC layer for creating personal area networks (PAN) with small, low-power digital radios in 900 MHz and 2.4 GHz. ZigBee even defines the network, security, and application framework for an IEEE 802.15.4-based system. Figure 1.3 illustrates the channels for IEEE 802.15.4 in 2.4 GHz frequency band. We dont consider that in 900 MHz.

Contrary to WiFi, ZigBee is typically used in low data rate applications that require long battery life. Therefore, signal from ZigBee device is designed with 10–100 times lower power compared with WiFi signal. As a result, ZigBee signal could be easily destructed by WiFi signal. Since only when WiFi and ZigBee work in the

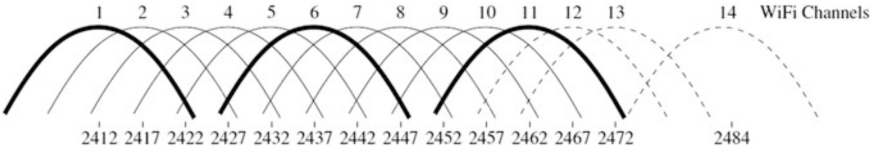


Fig. 1.2 Channel for 802.11 in 2.4 GHz

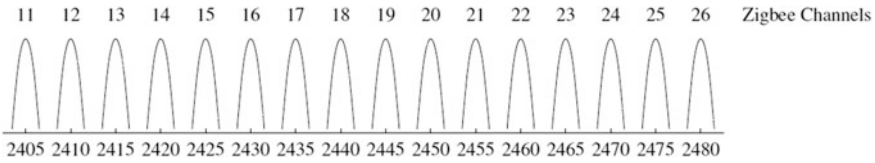


Fig. 1.3 Channel of ZigBee in 2.4 GHz

Table 1.1 Characteristic Parameters of WiFi and ZigBee

Technology	Baseband power (dbm)	Bandwidth (MHz)	Data rate (bps)
WiFi	0	20	2 M, 5 M, 10 M, 22 M
ZigBee	−20	2	250 K

same frequency band, interference will exist. Thus, in this book, we only discuss the interference problem in 2.4 GHz. Table 1.1 provides the detailed parameters for WiFi and ZigBee in terms of power, bandwidth and data rate.

1.3 Coexisting Scenarios of WiFi and ZigBee

Since WiFi networks have been deployed in every space in our life where there exists ZigBee-based system, the coexistence happens often. Here, we provide some examples to show the interference and coexistence problem between WiFi and ZigBee.

1.3.1 Smart Building

Although it has a variety of uses, ZigBee is perfect for commercial buildings, and it is one of the catalysts for smart building design because of its focus on dependability, long battery life, and simple operation. ZigBee technology has become the standard of choice among other wireless technologies by overcoming speed limitations found in other wireless commercial automation technologies and by allowing for more devices on a single network. The technology offers great energy efficiency via first-rate lighting, HVAC, and security control. ZigBee chips can be embedded in controllers, switches, and sensors and the savings for commercial facilities come from the cabling costs that are avoided. Because there’s no wiring involved, ZigBee’s peel-and-stick devices can be added anywhere—especially in locations where installing wired sensors would be difficult.

Since WiFi has been used as the main way of accessing internet for the resident in the building, the air space is filled with WiFi signal all the time. In this way, these control system will be interfered by WiFi signal and the corresponding solutions must be provided to handle the disruption from the cross-technology interference.

1.3.2 Wireless Body Area Networks

Wireless body area networks (WBANs) play a key role in future e-health. For example, one important WBAN application is multiparameter monitoring, where multiple vital signs of a patient are monitored continuously. These vital signs are sampled by the sensors mounted on the patient, and displayed on a central monitor.

Nowadays, more and more WBAN systems use ZigBee as the networking way for many sensors on the body and data transmission with the monitor. Therefore, all of them may suffer from existing interference from the pervasive WiFi networks, which run on the same 2.4 GHz band. Though the coexistence interference may not be a major concern for low duty cycle noncritical applications, such as body temperature monitoring, it is not the case for WBAN applications with stringent requirements on packet delivery ratio and/or latency.

However, a system of WBAN always works in an area with ubiquitous WiFi networks, and thus the coexistence in these systems cannot be ignored.

1.3.3 Heterogeneous Networking System

WiFi and ZigBee have different shortcoming and advantages in terms of communicating range, data rate, and energy consumptions. In some complex systems, the designers would like to choose WiFi technology to transmit data for a long distance and use ZigBee for sensing device to gather data. Thus, there are many systems are integrated with many kinds of networking technology. In these application scenarios, the working way of WiFi and ZigBee have to been carefully be designed and collaborated. With many such type of system deploying, the coexistence problem becomes more and more serious.

1.3.4 Signal Jamming

The simplicity of deployment and administration as well as low-cost hardware result in an increased reliance on wireless communication systems. However, the blocking of wireless communication, i.e., jamming, is one of the major security threats. In the context of wireless networks, jamming is the type of attack which interferes with the radio frequencies used by sensor nodes and may be viewed as a special case of denial of service (DoS) attacks. Especially, WiFi has a higher signal power than ZigBee and has the possibility to attack the low-power network by setting a high transmission power intentionally or unintentionally. In this case, ZigBee networks must consider how to work under jamming signal from WiFi.

1.4 A Glance at Interference Effect

In this section, we provide a glance at interference effects on ZigBee by experiments.

1.4.1 WiFi Interference on ZigBee

This experiment examines the effect of data rate variety when the distance is fixed. This experiment was done in the indoor environment which is the most likely to house overlapping WiFi and ZigBee networks. We chose the unusual WiFi channel 4 to minimize external interference and ensured that only our WiFi devices were working at channel 4 during the experiments. At the same time, ZigBee network is operated at channel 15 that is at the center of WiFi channel 4. In these experiments, the receiver reads RSSI register per 10 ms, and a total of 10,000 RSSI were collected for different WiFi interference rates.

Figure 1.4 plots the cumulative probability distribution (CDF) of RSSI sensed by ZigBee receiver under different WiFi interference rates. From the figure, significant distinction between different interferences could be easily observed. Especially, almost all (>95 %) RSSIs are lower than −45 dBm in the absence of interference.

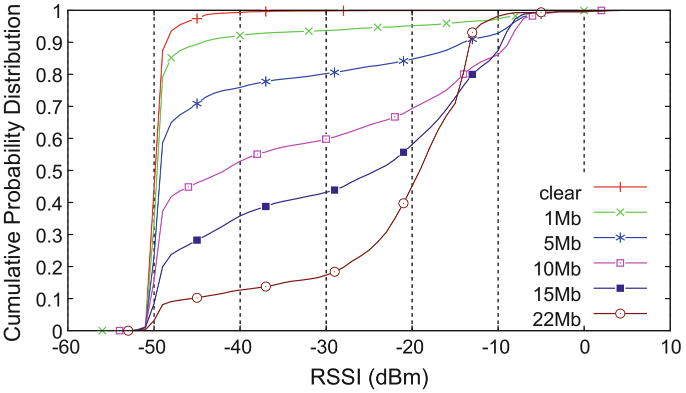


Fig. 1.4 The CDF of RSSI under different interference

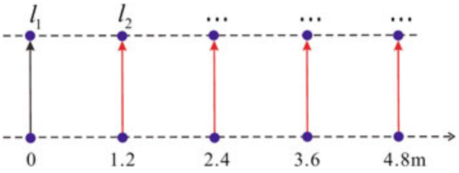


Fig. 1.5 Experiment configuration. Each link has a length (the distance from its sender to its receiver) of 3.6 m

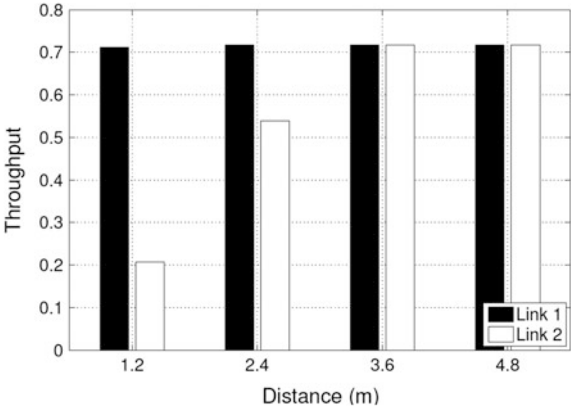


Fig. 1.6 The throughput of both links as functions of distance

1.4.2 ZigBee Interference on ZigBee

This experiment illustrates the interference in ZigBee network with different interference distances. The experiment topology is illustrated in Fig. 1.5. The experiment setting involves two parallel wireless links, l_1 and l_2, operated by four MicaZ nodes. The location of transmission link l_1 is fixed and the location of l_2 with the distance to l_1 is ranging from 1.2 to 4.8 m. The transmitters of the two links are made to send packets persistently as fast as possible, and the throughput of the two links, i.e., how many packets can be received correctly, is measured. The result is illustrated in Fig. 1.6, which shows that, interlink interference is varying with distance and the performance of one link will be suppressed if the distance is short. What is more, by increasing the distance between two links, both links may achieve a nearly full throughput.

1.5 Main Contents

In the following sections, we give the main contents of each chapter in this book.

1.5.1 Chapter 2 Fundamentals of WiFi and ZigBee

This chapter will introduce the fundamentals of WiFi and ZigBee, including the communication techniques from signal level, MAC protocol, networking architecture, and process. This chapter provides the foundation for discussing the following chapters.

1.5.2 Chapter 3 Interference Measurement and Model

This chapter will introduce the interference modes from the common communication theory and measurement statistics. These results have been reported in the related literature. This chapter will reorganize and classify these results so that the readers have deeper understanding on the interference.

1.5.3 Chapter 4 Interference Mitigation and Avoidance

This chapter will introduce the related techniques for interference mitigating and avoiding. The basic strategy is to arrange and adjust the working frequency of ZigBee or WiFi nodes dynamically.

1.5.4 Chapter 5 Coexistence Between WiFi and ZigBee

This chapter will introduce the coexisting techniques between WiFi and ZigBee networks. Coexistence means that WiFi and ZigBee could use the same working frequency and would not jam each other.

1.5.5 Chapter 6 Collaboration Between WiFi and ZigBee

This chapter will introduce how to collaborate the different wireless technology to meet some specific goal. The collaboration scenarios include energy-efficient AP discovery, heterogeneous radio communication, and so on.

Chapter 2
Fundamentals of ZigBee and WiFi

This chapter focuses on the fundamentals and technical characteristics which will bring up significant effects on the interference and solutions. Since it is much easier for ZigBee to be interfered with by WiFi, this chapter will discuss ZigBee more.

2.1 ZigBee

Based on IEEE 802.15.4, ZigBee is currently the de facto standard for wireless sensor networks (WSNs) [1, 2]. It is designed for ZigBee focuses on the field of low-power, low-cost, and low-bit rate communications, which has been widely used in sensor networks, cyber-physical systems [4], and smart buildings. We will first introduce the basic conceptions in ZigBee networks and then introduce some important mechanisms in ZigBee specification. These mechanisms will be exploited for handling interference from WiFi in the following chapters.

2.1.1 Overview of ZigBee

ZigBee is a specification that is built on top of the IEEE 802.15.4 short-range communications standard [3]. The name ZigBee comes from the fact that bees can dance to pass messages to each other, also in a multihop fashion. ZigBee covers the upper layers of the protocol stack, while 802.15.4 is in charge of MAC and PHY layers.

ZigBee is intended for low-throughput, low-power, low-cost applications. For this reason, it is much simpler than other protocols such as WiFi (IEEE 802.11). It has support for mesh topologies, which means that ZigBee devices relay messages for each other through multiple wireless hops.

G. Shi and K. Li, *Signal Interference in WiFi and ZigBee Networks*,
Wireless Networks, DOI 10.1007/978-3-319-47806-7_2

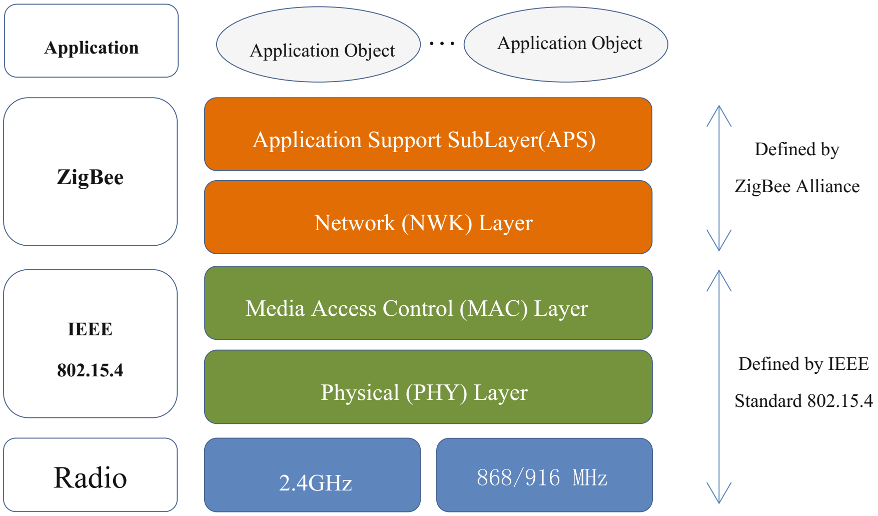


Fig. 2.1 ZigBee protocol stack architecture

Figure 2.1 depicts ZigBee protocol stack, which consists of four layers, viz., PHY, MAC, network and application layer. The first two are covered in IEEE 802.15.4 standard and the latter two are covered in documents published by ZigBee alliance. The first two are the critical factors for handling the interference from WiFi. Therefore, the following content focuses on MAC and PHY layer.

There are three network topologies in ZigBee. Besides the star topology, the ZigBee network layer also supports more complex topologies like the tree and the mesh, shown in the image below. Among the functionalities provided by the network layer are multihop routing, route discovery and maintenance, security and joining/leaving a network, with consequent short (16-bit) address assignment to newly joined devices.

There are three main players in a ZigBee network:

- **Coordinator**: is the most powerful device. There is a single coordinator in each network. It is the node that creates the network and the other nodes simply join in. Quite often, this is the sink of the WSN, which gathers all the data that is transmitted. One of the coordinator tasks is to assign short addresses.
- **Router**: are intermediate devices. They can relay packets for other nodes. They join a network that already exists and then announce it using beacons. Therefore, they can have "children" nodes that join the network by establishing communication with the router.
- **End Devices**: these are the simplest devices. They cannot forward packets nor have children that depend on them and, quite often, they enter a sleep mode in order to save energy. Figure 2.2 illustrates the players in three different network topologies which are Star, Tree and Mesh, respectively.

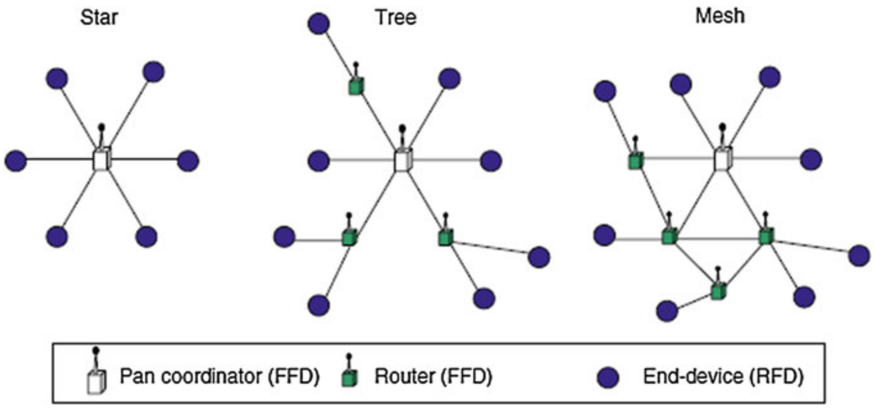


Fig. 2.2 Network topology

There are two types of data transfer transactions in ZigBee networks.

- The first one is the data transfer between a coordinator and a device, in which a device transmits the data to or receives the data from a coordinator. This transaction is used in star topology.
- The second transaction is the data transfer between two peer devices. In a peer-to-peer topology, data is exchanged between any two devices on the network; consequently all three transactions are used in this topology.

2.1.2 IEEE 802.15.4 Physical Layer

Since the interference effect on ZigBee is more prominent than that on WiFi, many works search solutions from physical layer. Thus, in this subsection, we will provide the key designs in ZigBee physical layer.

The PHY provides an interface between the MAC sublayer and the physical radio channel, via the RF firmware and the RF hardware. Besides radio on/off operation, the physical layer includes functionalities of channel selection, link quality estimation, energy detection (ED) measurement, and clear channel assessment (CCA).

2.1.2.1 Channel Assignment and Switching

The IEEE 802.15.4 PHY layer supports three frequency bands: a 2450 MHz band (with 16 channels), a 915 MHz band (with 10 channels), and an 868 MHz band (1 channel), all using the direct sequence spread spectrum (DSSS) access mode. In these three frequency bands, only 2.4 GHz band overlaps with that of WiFi. Thus, we only introduce the protocol defined for operating in the 2.4 GHz ISM band.

The frequency band for IEEE 802.15.4 in 2.4 GHz ranges from 2400 to 2483.5 MHz. Furthermore, these spectrums are subdivided into channels with a center frequency and bandwidth. It is well known that the IEEE 802.15.4 standard defines 16 channels within this band, each 2 MHz wide with 3 MHz interchannel gap-bands (see Fig. 2.3). The center frequency of these channels can be calculated as follows:

$$F_c = 2405 + 5(k - 11) \text{ in megahertz, for } \quad k = 11, 12, \ldots, 26$$

where F_c and k are the center frequency and channel number, respectively. The detailed frequency ranges and the center frequencies of every channel are listed in Table 2.1.

Although these channels are nonoverlapping in frequency band, a channel is not orthogonal to all the other channels. Some results show that concurrent transmissions on adjacent channels will result in interference [4]. This is due to energy spill over and imperfect filtering. In comparison, the two channel away interference will

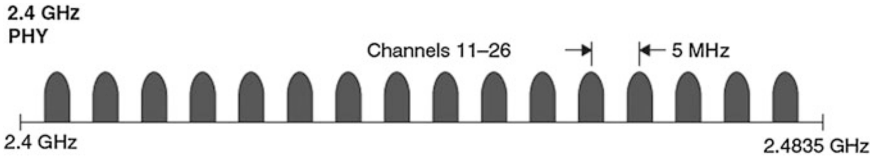


Fig. 2.3 Frequency and channel for ZigBee in 2.4 GHz

Table 2.1 GHz PHY channel frequencies

Channel ID	Lower frequency	Center frequency	Upper frequency
11	2404	2405	2406
12	2409	2410	2411
13	2414	2415	2416
14	2419	2420	2421
15	2424	2425	2426
16	2429	2430	2431
17	2434	2435	2436
18	2439	2440	2441
19	2444	2445	2446
20	2449	2450	2451
21	2454	2455	2456
22	2459	2460	2461
23	2464	2465	2466
24	2469	2470	2471
25	2474	2475	2476
26	2479	2480	2481

be smaller. Therefore, the 16 channels can however be divided into two sets of orthogonal 3 channels, with each containing 8 channels—(11, 13,…, 25) and (12, 14,…, 26). *More interference models and experiments will be discussed in* Chap. 3.

Generally, the working channel of ZigBee network is predefined. But all devices can trigger scanning operations for dynamic channel selection. Channels are scanned in order from the lowest channel number to the highest if the scanning is for channel selection. The scanning process will provide the energy level feedback and the nodes will select the quietest one for their new working channel.

The channel switching operation, via channel register writing, can only happen when the radio is in IDLE state and will induce cost of time. In another word, the channel switching will not come into effect immediately if the request is sent out when the radio is not in IDLE state. The procedures for channel switching roughly include radio status change, channel number register writing, and PLL (phase-locked loop) calibrating. In literature [5], the experiments with Micaz mots show that the time to switch between channels and wait until the frequency synthesizer stabilizes is roughly equal to the time to transmit one packet with 32 bytes. The widely used CC2420 transceiver has channel switching times of only 300 microseconds. More recent radios have dramatically reduced the time it takes to switch channels. For example, the newer Chipcon CC2500 2.4 GHz radio takes only 90 μs to switch channel. But this time is measured when the radio has been calibrated at startup [6]. Therefore changing to another frequency channel dynamically and frequently has the potential to become a damper on system performance.

2.1.2.2 Energy Detection and CCA

The receiver ED measurement in IEEE 802.15.4 is intended for use by a network layer as part of channel selection algorithm. It is an estimate of the received signal power within the bandwidth of an IEEE 802.15.4 channel. No attempt is made to identify or decode signals on the channel. The ED time should be equal to 8 symbol periods. Generally, the ED value is calculated by averaging RSSI values over eight symbols (128 μs).

The ED result shows the power level of received signal including interference and noise. The ZigBee network node could use this information to infer the interference condition so that a better channel could be selected.

The ED report defined in the standard is an 8-bit integer ranging from 0x00 to 0xff. The minimum ED value (0) shall indicate received power less than 10 dB above the specified receiver sensitivity. The range of received power spanned by the ED values shall be at least 40 dB. Within this range, the mapping from the received power in decibels to ED values shall be linear with an accuracy of ±6 dB.

CCA, partially based on ED, is an essential ingredient in wireless networks employing channel sensing as part of their medium access mechanism. CCA is implemented at the PHY layer from the view of protocol stack, but it is often used by MAC layer. When the MAC layer receives a packet to transmit, it instructs the PHY to do CCA as described in the following MAC section.

The standard specifies that the CCA duration shall be 8 symbol periods or 128 μs. The CCA is performed according to at least one of the following three methods:

- Energy above threshold. CCA shall report a busy medium upon detecting any energy above the ED threshold.
- CS only. CCA shall report a busy medium only upon the detection of a signal with the modulation and spreading characteristics of IEEE 802.15.4. This signal may be above or below the ED threshold.
- Carrier sense with energy above threshold. CCA shall report a busy medium only upon the detection of a signal with the modulation and spreading characteristics of IEEE 802.15.4 with energy above the ED threshold.

Take CC2420, a popular ZigBee-compliant RF transceiver, as an example, it has a built-in RSSI (received signal strength indicator) which has the likewise effect of ED and provides a digital value that can be read from the 8 bit, signed 2's complement RSSI.RSSI_VAL register. As defined in IEEE 802.15.4 standard, the RSSI value is always averaged over 8 symbol periods (128 μs). The RSSI register value RSSI.RSSI_VAL can be referred to the power P at the RF pins by using the following equations:

$$P = \mathrm{RSSI\_VAL} + \mathrm{RSSI\_OFFSET}\ [\mathrm{dBm}]$$

where the RSSI_OFFSET is found empirically during system development from the front-end gain. RSSI_OFFSET is approximately −45. For example, while reading a value of −20 from the RSSI register, the RF input power is approximately −65 dBm.

2.1.2.3 PHY Protocol Data Units

The PHY protocol data unit is called PPDU, which encloses the MAC frames passed to the PHY as the PHY service data unit (PSDU).

The schematic view of PPDU is illustrated in Fig. 2.4. Each PPDU packet consists of the following basic components:

- SHR, which allows a receiving device to synchronize and lock into the bit stream.
- PHR, which contains frame length information.
- PSDU, a variable length payload, which carries the MAC sublayer frame.

The PPDU packet structure is illustrated in Fig. 2.5. PPDU begins with a preamble sequence which is composed of 32 zeros (all bytes set to 0x00) and is used for chip and symbol synchronization at receiver part of transceiver. Start of frame delimiter (SFD) follows the preamble and is 8 bit field segregate between

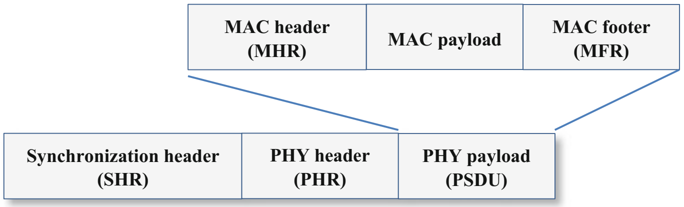


Fig. 2.4 Schematic view of the PPDU

preamble and actual physical layer data. The SFD indicates the end of the SHR and the start of the packet data and is set to 0x7A. At the receiver side, an 802.15.4 radio synchronizes to incoming zero-symbols and searches for the SFD sequence to receive incoming packets. The PHR includes a 1-byte length field that describes the number of bytes in the packet's payload, including the 2-byte CRC. PSDU field carries PHY packet but the payload is transferred from MAC sublayer. Its length is variable.

From Fig. 2.5 we can see that ZigBee PPDU Frame consists of SHR (Preamble 4 bytes, SFD 1 byte) + PHR (Frame length 1 bit, Reserved 1 bit) + PHY Payload (PSDU variable length). Therefore, *the maximum packet size in IEEE 802.15.4 is 133 bytes*, including all the headers. Note that while the IEEE 802.15.4 specification mandates a 4-byte preamble, some radios such as the CC2420 allow the user to set the length of the transmitted preamble up to 17 bytes.

2.1.2.4 Modulation and Spreading

Any information to be transmitted via ZigBee must be modulated firstly. To improve signal-to-noise Ratio (SNR) of received signals at the receiver, ZigBee employs direct sequence spread spectrum (DSSS) that uses a digital spreading function representing pseudorandom noise (PN) chip sequences [1]. A bit in a PN-code is called a *chip* and thus a PN-code can be also called a *chip sequence*. In reality, each symbol is presented by using predefined chip sequences.

At the sender side before the bit sequences are being modulated and transmitted through the antenna, there is an additional process to chop the sequences into symbols and replace each symbol with the corresponding chip sequence, which is modulated to baseband transmission waveform and is ultimately transmitted over the air.

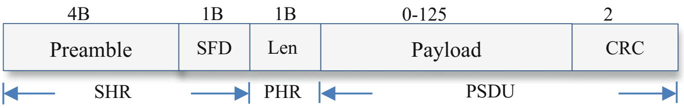


Fig. 2.5 802.15.4 PHY protocol data units format

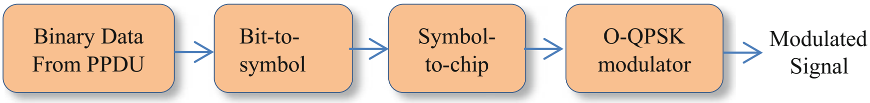


Fig. 2.6 Modulation and spreading functions for the O-QPSK PHYs

To do this, outgoing bytes are divided into two 4-bit symbols, the 4 least significant bits (LSB) and the 4 most significant bits (MSB). Each 4-bit symbol will be spread to a specified 32-bit long PN sequence. IEEE 802.15.4 predefines the map table from 4-bits symbol to 32-bits chip sequences, as illustrated in Table 2.2. The radio encodes these chip sequences using orthogonal quadrature phase shift keying (O-QPSK) and transmits them at 2 Mchips/s (i.e., 250 kbps). O-QPSK PHY is mandatory when IEEE 802.15.4 is operating in the 2450 MHz band. The modulating and spreading process in IEEE 802.15.4 PHY is illustrated in Fig. 2.6.

For example, one byte binary data from PPDU, denoted as $(b_0b_1b_2b_3b_4b_5b_6b_7)$, is first grouped into two nibbles of 4-bit symbols $(b_0b_1b_2b_3)$ and $(b_4b_5b_6b_7)$. And then each 4-bit symbol will be spread to a specified 32-bit long PN sequence $\leftarrow C_0C_1C_2 \ldots C_{31}$ which is also called chips sequence. Each bit or chip (C_i) in a PN sequence is then modulated using O-QPSK. The modulated O-QPSK signal goes to the half-sine pulse shaping stage and then the digital–analog conversion converts the digital baseband waveform into the analog baseband waveform. The radio front-end up-converts the baseband waveform to 2.4 GHz carrier and transmits it by the radio frequency (RF) transmitter finally [7].

Specifically, the even chips $C_0C_2C_4\ldots$ are modulated as In-phase (I) component of the carrier and the odd indexed chips $C_1C_3C_5\ldots$ are modulated as Quadrature

Table 2.2 Symbol-to-chip mapping table

Data symbol (b_0 b_1 b_2 b_3)	Chip values (c_0 c_1 … c_{30} c_{31})
0000	(PN_1) = 11011001110000110101001000101110
1000	(PN_2) = 11101101100111000011010100100010
0100	(PN_3) = 00101110110110011100001101010010
1100	(PN_4) = 00100010111011011001110000110101
0010	(PN_5) = 01010010001011101101100111000011
1010	(PN_6) = 00110101001000101110110110011100
0110	(PN_7) = 11000011010100100010111011011001
1110	(PN_8) = 10011100001101010010001011101101
0001	(PN_9) = 10001100100101100000011101111011
1001	(PN_{10}) = 10111000110010010110000001110111
0101	(PN_{11}) = 01111011100011001001011000000111
1101	(PN_{12}) = 01110111101110001100100101100000
0011	(PN_{13}) = 00000111011110111000110010010110
1011	(PN_{14}) = 01100000011101111011100011001001
0111	(PN_{15}) = 10010110000001110111101110001100
1111	(PN_{16}) = 11001001011000000111011110111000

(Q) component of the carrier. A chip ‘1’ is shaped to a positive half-sine and a chip ‘0’ is shaped to a negative half-sine as shown in Fig. 2.7. Here ‘O’ in O-QPSK expresses a half chip time offset. Since the time duration of each chip is 1 μs, the time offset between the Q-phase chips and I-phase chips is a half chip time, i.e., 1 μs/2 = 0.5 μs which is illustrated in Fig. 2.7. This offset results in a continuous phase change and constant envelope. For more implementation information about O-QPSK, please refer to literature [8].

Based above introduction, Fig. 2.8 summarizes the whole process of spectrum spreading in IEEE 802.15.4. For demodulation, the receiver’s radio converts each half-sine pulse signal into a chip. Then these chips are grouped to provide PN sequences. The de-spreading is performed by mapping the PN sequence to the symbol with the highest correlation. A correlator is responsible for separating PN-Codes out of all chips that were received. The correlator captures chip sequences that are the same or similar to PN-Codes defined in Table 2.2. It then tries to find a best-match PN-Code for a chip sequence.

In the ideal case, the “best-match” PN-Code should be exactly equal to the captured chip sequence; whereas, in real situations, it is a different story. Although a sender should never transmit a wrong PN-Code (a chip sequence not included in the predefined Table), some chips could be corrupted during transmission in the presence of interference and multipath. Interference and noise can corrupt the incoming chip stream, leading to 32-chip sequences that do not match one of the 16 valid sequences. In case of corrupted chips, however, the “best-match” PN-Code need not fully agree with the erroneous chip sequence.

There are various methods to find out a “best-match” PN-Code. One such method is maximum likelihood decoder (MLD) where each received 32-bit chip sequence P is compared with the predefined PN-Codes PN_1; PN_2; … ; PN_{16} in Table 2.2 in turn to find out the corresponding symbol such that the *hamming distance* of P and the PN-code of the symbol are minimized. Here *hamming distance* is the number of different positions of two bit strings. In case of corrupted packet, the receiver maps the input sequence to the valid sequence with the smallest Hamming distance.

Besides, the literature [9] mentions that some 802.15.4 radios (e.g., CC2420) enable users to control the correlation threshold to control the maximum Hamming distance between the received 32-chip sequence and the valid SFD sequence that the receiver is willing to tolerate. If this threshold is high, the received signal must closely match the ideal signal. If this threshold is low, the receiver allows a low signal-to-noise ratio at the expense of potentially interpreting corrupted packets or channel noise as valid packets.

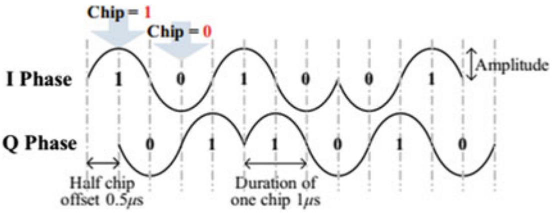
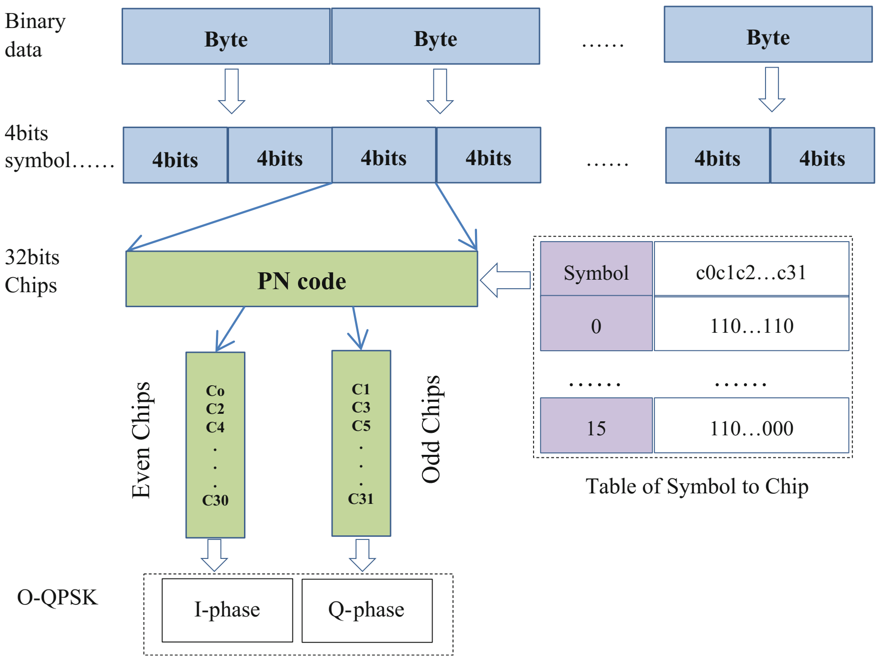


Fig. 2.7 Half-sine pulse shaping in O-QPSK



Symbol	c0c1c2…c31
0	110…110
……	……
15	110…000

Fig. 2.8 The flowchart of spectrum spreading

2.1.3 *IEEE 802.15.4 MAC Layer*

The MAC layer in IEEE 802.15.4 handles all access to the physical radio channel and is responsible for the tasks such as generating network beacons if the device is a coordinator, supporting PAN association, employing the CSMA-CA mechanism for channel access, and so on. We have no plan to introduce every detail of IEEE 802.15.4 MAC Layer. We only focus on the related protocol procedures and characteristics with the methods of interference handling.

2.1.3.1 MAC Frame Format

In the PHY protocol data unit (PPDU), the PSDU is enclosed following PHY header PHR and contains the MAC Header (MHR), which has two frame control octets, a single octet data sequence number, good for reassembling packets received out of sequence, and 4–20 octets of address data. The MAC service data unit (MSDU) carries the frame’s payload and has a maximum capacity of 104 octets of data. Finally, the MPDU ends with the MAC footer (MFR), which contains a 16-bit frame check sequence (FCS). The frame format is illustrated in Fig. 2.9.

Table 2.3 summarizes the bit length of each field in PPDU.

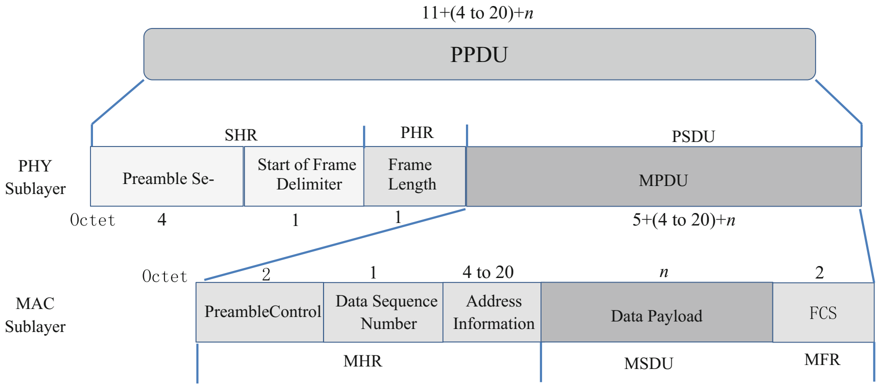


Fig. 2.9 MAC frame format and the layout in PPDU

Table 2.3 Bit length of PPDU

Field	Bit length
Preamble sequence	32
SFD	8
PHR	8
Frame control	16
Sequence number	8
FCS	16

2.1.3.2 Channel Access

An IEEE 802.15.4 network can work either in beacon-enabled or in non-Beacon mode. In the beacon-enabled mode, a network coordinator transmits regular beacons for synchronization and association procedures to control communication. Data transfer between a device and a coordinator is synchronized in a superframe. The superframe can have an active and an inactive portion. All communications take place in the active period while nodes are allowed to enter a low-power mode during the inactive period.

Furthermore, the active period in turn may consist of a contention access period (CAP) and a contention-free period (CFP). Channel access in the CAP is in the form of slotted CSMA/CA for contention access. Meanwhile, the guaranteed time slot (GTS) forms the CFP which is dedicated to low-latency applications or applications requiring specific data bandwidth. CFP always appears at the end of the active superframe starting at a slot boundary immediately following the CAP. The detailed superframe structure can be found in the IEEE 802.15.4 specification.

In the beaconless mode, there are no regular beacons, and devices communicate with each other using unslotted CSMA/CA protocol for channel access. According to the IEEE 802.15.4 protocol, the unslotted CSMA/CA algorithm is similar to the slotted CSMA/CA algorithm, besides that it is used in the non-Beacon-enabled mode. In the ***unslotted CSMA/CA***, each time a device wishes to transmit data frames or MAC commands, it waits for a random period. If the channel is found to be idle (done by CCA mechanism), following the random back-off, the device transmits its data. If the channel is found to be busy following the random back-off, the device waits for another random period before trying to access the channel again. Acknowledgment frames are sent without using a CSMA-CA mechanism.

When ***slotted CSMA/CA*** is employed, the back-off periods of one device are aligned with the start of the beacon transmission. Each time a device wishes to transmit data frames during the CAP, it locates the boundary of the next back-off period and then waits for a random number of back-off periods. If the channel is idle, the device begins transmitting on the next available back-off period boundary. In conclusion, the status switching in slotted CSMA/CA should be related with the slots arrangement.

While the *classical CSMA/CA* protocol uses binary exponential back-off, in practice some CSMA/CA protocol implemented in TinyOS uses a fixed length back-off interval [10]. At the same time, IEEE 802.15.4 does not employ RTS/CTS since the normal packet has a short packet length, compared with IEEE 802.11. This RTS/CTS overhead proves to be useful when traffic load is high, but obviously too expensive for low-data rate applications as of the case of WPANs for which IEEE 802.15.4 is designed.

2.2 WiFi

2.2.1 Overview of WiFi

Wireless fidelity (WiFi) includes IEEE 802.11 a/b/g standards for wireless local area networks (WLAN) [11, 12]. It is designed to enable users to surf the Internet at broadband speeds with mobile wireless devices via an access point (AP) or in ad hoc mode. The IEEE 802.11 architecture consists of several components that interact to provide a wireless LAN that supports station mobility transparently to upper layers. Since we are focusing on the signal interference from WiFi, we do not introduce all functions such as the mobility mechanism supported by WiFi.

The IEEE 802.11 standard covers both the medium access control (MAC) and physical (PHY) layers. These two layers have direct effect on the interference characteristics. Like in Sect. 2.1, the following content manifests the key protocol components in these two layers.

2.2.2 IEEE 802.11 Physical Layer

The original 802.11 standard defines a DSSS system operating in the 2.4 GHz frequency band. A number of amendments have greatly expanded WLAN capability by specifying more modulation and coding schemes and more frequency bands.

However, IEEE 802.11a works in 5 GHz ISM band, it does not have interference effect on ZigBee in 2.4 GHz. Thus, we only consider the standard that works in 2.4 GHz, including IEEE 802.11b/g/n. IEEE 802.11b is the first to reach mass production, which runs DSSS in the 2.4 GHz ISM band and 802.11g are orthogonal frequency division multiplexing (OFDM) systems. IEEE 802.11n mainly enhances the previous three by adding multiple-input multiple-output (MIMO) antenna support.

- Frequency Occupation and Channel

Every WiFi subtype standard predefines a fixed set of RF channels. Though a single WiFi network can only use one of these predefined RF channels, when several WiFi networks coexist in an area, they will try or will be configured to use nonoverlapping RF channels. This can easily exhaust the whole 2.4 GHz ISM band. For example, two coexisting IEEE 802.11n networks, each with 20 MHz frequency width, are enough to occupy the whole 2.4 GHz ISM band as illustrated in Fig. 2.10. Such scenario is not uncommon nowadays given the ubiquitous presence of WiFi networks. When all such WiFi networks are active, jamming the whole 2.4 GHz ISM band, it is hard to carry out WBAN communications, no matter the WBAN uses ZigBee, Bluetooth, or the draft IEEE 802.15.6 2.4 GHz standard (Fig. 2.10).

- DSSS and OFDM

DSSS and OFDM is the two most used RF transmission techniques in IEEE 802.11 standard. DSSS is also used in ZigBee. The fundamentals are the same with that in ZigBee but the spreading processes have serval differences. In IEEE 802.11, a single PN-code is used by every user in the network. This PN-code is the 11 bit barker sequence: +1 −1 +1 +1 −1 +1 +1 +1 −1 −1 −1. Spectrum spreading of each bit can be thought of as XORing operation on a stream of data bits with this specific PN sequence. As result, a “one” or a “zero” is transmitted as 11 bits of data represented by the original Barker sequence or the inverse of the Barker sequence. However, in ZigBee, there are 16 32-bit long PN-codes corresponding to 4-bit symbol. The spectrum spreading is done through predefined mapping.

Different with DSSS, OFDM is a multi-carrier modulation scheme that extends the concept of single subcarrier modulation by using multiple subcarriers within the same single channel. OFDM divides the used RF bandwidth into many narrow sub-channels called OFDM bins or subcarrier. Each OFDM bin can be treated

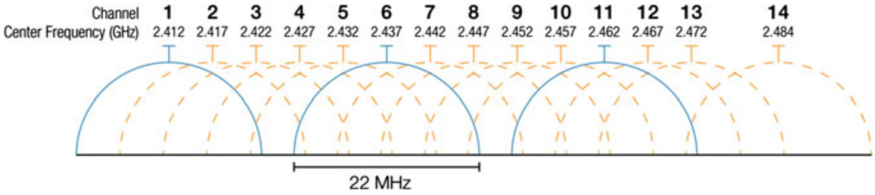


Fig. 2.10 Frequency and channel for IEEE 802.11 in 2.4 GHz

independently from other bins, and may use a different modulation (e.g., BPSK, 4-QAM) or transmission power. For 802.11 in 2.4 GHz, there are 52 OFDM subcarriers, 48 are for data.

In OFDM, a data stream is striped into bits, with different numbers of bits assigned to each bin based on its modulation scheme. An assignment of modulated bits to each of the OFDM bins is called an OFDM symbol, see Fig. 2.11. The frequency domain OFDM symbol is converted to a time domain OFDM symbol by using an inverse fast Fourier transform (IFFT) and sent on the medium by the transmitter. The receiver passes the received signal to a fast Fourier transform (FFT) module to produce the frequency representation. The data symbols are then converted to their frequency representation, corrected for the channel, and demodulated to retrieve the transmitted data bits.

- PHY Packet Format

Due to backward compatibility considerations, all subtypes of WiFi running in 2.4 GHz ISM band recognize the IEEE 802.11b packet format.

Viewing from PHY layer, a WiFi packet transmission begins with a PHY preamble, followed by a PHY header, and then the DATA. The PHY encapsulation for IEEE 802.11 is illustrated in Fig. 2.12. The PHY preamble is for receiver carrier acquisition which is a DSSS modulated signal. The PHY header contains several fields that carry control/management information. *LENGTH* field is a 16-bit unsigned integer specifying the number of microseconds that WiFi packet lasts. This implies that a maximum of 65,535 μs can be reserved for DATA segment.

Other fields in PHY frame includes:

- SYNC. This field consists of alternating 0 s and 1 s, alerting the receiver that a receivable signal is present. The receiver begins synchronizing with the incoming signal after detecting the YNC.
- Start Frame Delimiter. This field is always 1111001110100000 and defines the beginning of a frame.
- SIGNAL. This field identifies the data rate of the 802.11 frame, with its binary value equal to the data rate divided by 100 Kbps. For example, the field contains the value of 00001010 for 1 Mbps, 00010100 for 2 Mbps, and so on. The PLCP

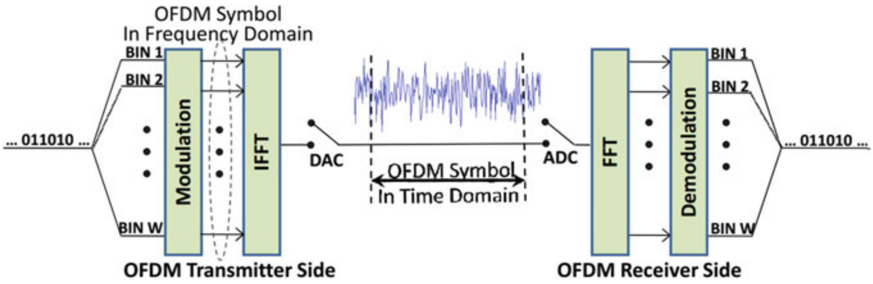


Fig. 2.11 Schematic of an OFDM system [13]

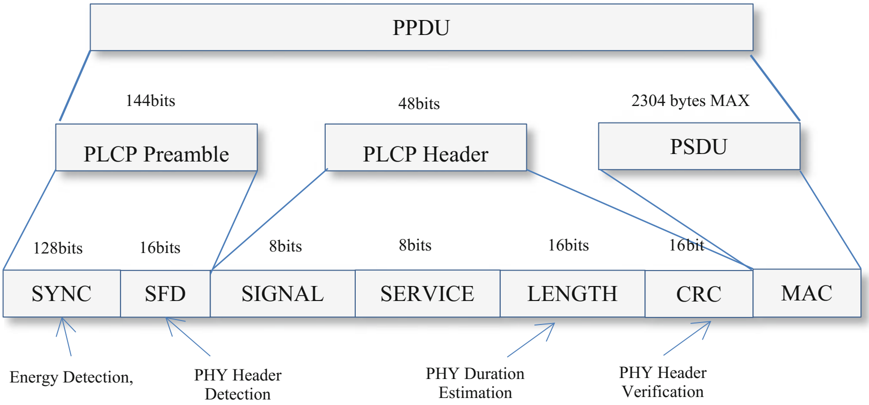


Fig. 2.12 PHY encapsulation for IEEE 802.11

fields, however, are always sent at the lowest rate, which is 1 Mbps. This ensures that the receiver initially uses the correct demodulation mechanism, which changes with different data rates.

- Service. This field is always set to 00000000, and the 802.11 standard reserves it for future use.
- Frame Check Sequence. In order to detect possible errors in the Physical Layer header, the standard defines this field for containing 16-bit cyclic redundancy check (CRC) result. The MAC Layer also performs error detection functions on the PPDU contents as well.
- PSDU. The PSDU, which stands for Physical Layer Service Data Unit, is a fancy name that represents the contents of the PPDU (i.e., the actual 802.11 frame being sent).

- Clear Channel Assessment

IEEE 802.11 standard also have the necessity function of CCA. Like ZigBee, all subtypes of WiFi carry out carrier sense multiple access (CSMA) MAC protocol. An IEEE 802.11 node shall always listen to the wireless medium before transmission. Only when the wireless medium is idle the node start transmitting. This procedure is called CCA.

The CCA mechanism used in IEEE 802.11 has the same working way with 802.15.4. There are also three types of CCA: ED only CCA measures the wireless medium spectral power level; if it is greater than a threshold, the wireless medium is considered busy. Carrier sense (CS) only CCA tries to capture WiFi PHY preambles; if a PHY preamble is successfully captured, the wireless medium is considered busy. Usually, CS-only CCA also looks into the content of the PHY header immediately following the captured PHY preamble (if there is one) to provide more accurate CCA evaluations. ED + CS CCA does both. In practice, CS-only CCA and ED + CS CCA are most widely implemented.

Under the constraint of CCA, a node having packet to be sent enters the transmit mode and waits for a certain time period to make sure the medium is free (CSMA). The determination process is done by using the CCA module that may be configured in above three modes to make this determination. Only when the result is true, the node will have the probability to transmit the packets.

- Reception Handling in PHY Layer

Reception at a node can be explained in terms of the PLCP (physical layer convergence protocol) headers that encapsulate packets (shown in Fig. 2.11).

At the receiver side, the SYNC binary data will be first detected when it is alerting with 0, 1 sequence mode. A preamble of a *SYNC* bit pattern will trigger the ED circuitry that alerts the receiver to an incoming transmission. This 0, 1 altering bit pattern is also used to extract symbol timing. It is always transmitted at 1 Mbps. 802.11b/g uses either a long preamble that transmits the PLCP header (Fig. 2.11) at 1 Mbps or a short preamble that transmits the PLCP header at 2 Mbps, regardless of the transmit speed of the MAC frame itself.

The receive procedure is invoked by the CCA procedure upon detecting a portion of the preamble sync pattern followed by a valid SFD and PLCP Header. The SFD is a specific 16-bit pattern (0x07cf with long preambles) that signifies the start of PLCP data.

After a WiFi device detects a PHY preamble and decodes the following PHY header, it will mute (i.e., refrain from transmitting) for a number of microseconds depending on the received *LENGTH* field and the device's specific implementation. The *LENGTH* field defines the packet length, which is used with bit rate information in the *SERVICE* field to determine the overall duration of the packet. To complete the PLCP processing, the receiver computes a CRC over the header. It generates a physical layer error if the header is corrupted. The MAC frame follows and it includes a separate CRC over the MAC contents. The receiver generates a separate MAC layer error if the MAC is corrupted. Otherwise, it will deliver the received packets to MAC layer.

2.2.3 IEEE 802.11 MAC Layer

Like IEEE 802.15.4, IEEE 802.11 also has two fundamental modes: distributed coordination function (DCF) and point coordination function (PCF). In DCF mode, each station must sense the status of the wireless medium before transmitting. However, in PCF, a point coordinator also known as AP, coordinates the communication. Both modes are based on CSMA/CA mechanism. The 802.11 standard specifies using CSMA/CA with ACKs as the MAC protocol, optionally with the

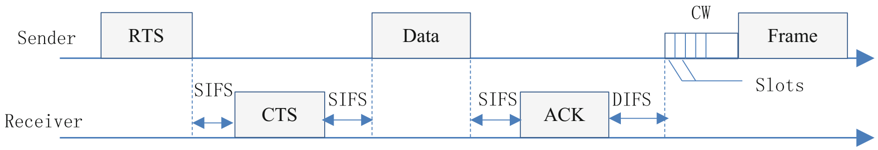


Fig. 2.13 Timing diagram for CSMA/CA with RTS/CTS

addition of RTS/CTS packets. The protocol also specifies the SIFS (short interframe space)[1] and DIFS (DCF interframe space)[2] intervals when nodes should defer using the medium.

Figure 2.13 presents the key features of the 802.11 MAC protocol through a timing diagram. In IEEE 802.11, the CCA module introduced in Sect. 2.2.2 is used to detect whether the medium is free and if it declares the medium to be free, the packet is sent. If it is busy, the transmitter defers the transmission for a random number of 20 μs slots selected between 1 and the contention window (CW), and repeats the CCA procedure.

The CW following the DIFS is shown in the figure. This window is divided into slots and is doubled every time they fail to access the medium, until CW reaches a maximum size of 1023 slots; the packet is sent if this maximum is reached regardless of whether the medium is busy. Nodes use a uniform random distribution to select a slot and wait for that slot before attempting to access the medium. The node that selects the earliest slot wins while others defer. The CW is reset to a minimum value 31 slots after a transmission.

When receivers receive a nonbroadcast data packet that passes the CRC check for data integrity, they send an ACK packet within a fixed time limit to acknowledge the receipt. If the transmitter does not receive an ACK, it considers the packet (or its ACK) lost. It then retransmits the packet by reinserting it at the front of the transmission queue and treating it as a new packet. Retransmission can be repeated up to seven times, after which the packet is dropped. Optionally, nodes can precede data packets with a RTS/CTS exchange to reduce the likelihood of interference by hidden terminals, but most implementations choose not to do so in practice because the costs outweigh the benefits.

Table 2.4 summarizes the duration of the DIFS, SIFS, and backoff slots for 802.11b and 802.11g. Also shown are the maximum and minimum packet sizes for 802.11b, 802.11g, and 802.15.4. It is worth noting that for many 802.11b and 802.11g packets, the entire air time is smaller than an 802.15.4 slot time. Based on the relatively small time intervals between 802.11 transmissions, one can easily see that a backlogged 802.11 sender can potentially corrupt the vast majority of 802.15.4 packets.

[1] SIFS is the amount of time in micro seconds required for a wireless interface to process a received frame and to respond with a response frame.

[2] If a node finds that the medium is continuously idle for DCF interframe space (DIFS) duration, it is then permitted to transmit a frame.

Table 2.4 Packet and interval durations for 802.11 and 802.15.4 [9]

Parameter	802.15.4	802.11b	802.11g
SIFS	N/A	30 μs	10 μs
DIFS	N/A	50 μs	28 μs
Slot time	320 μs	20 μs	9 μs
Initial CW	1–32	0–31	0–31
Successive CWs	1–8	BEB	BEB
Min length packet	352 μs	202 μs	194 μs
Max length packet	4256 μs	1906 μs	542 μs

Table 2.5 Features comparison between WiFi and ZigBee

Standard	ZigBee	WiFi
IEEE Spec	802.15.4	802.11b/g/n
Frequency band	2.4 GHz	2.4 GHz
Max signal rate	250 kbps	54 Mbps
Nominal range	10–100 m	100 m
Nominal TX power	(−25)–0 dBm	15–20 dBm
Number of channels	16	14
Channel bandwidth	2 MHz	22 MHz
Spreading	DSSS	DSSS, OFDM
Data protection	16-bit CRC	32-bit CRC

Besides above characteristics, the 802.11 MAC also defines management packets, the most relevant here being beacons and probes. An AP periodically (~100 ms) broadcasts beacons to assist clients with association, roaming, synchronization, power saving and other tasks. Beacons carry an 8-octet timestamp field so that the client's NIC can synchronize its clock with the AP to meet the timing constraints of the 802.11 MAC. Probe packets are sent by a client to discover APs.

2.3 Summary

In this chapter, we introduce the key design features in IEEE 802.11 and IEEE 802.15.4. These issues are directly related with the forthcoming solutions discussed in the following content. Here we provide the comparisons in protocol features between WiFi and ZigBee as the summaries in Table 2.5.

References

1. IEEE Computer Society, 802.15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). Available at: http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf
2. J. Eidson, E.A. Lee, S. Matic, S.A. Seshia, J. Zou, Distributed real-time software for cyber-physical systems. Proc. IEEE (special issue on CPS) **100**(1), 45–59 (2012)

3. ZigBee Alliance, ZigBee specification. ZigBee document 053474r17, (2008)
4. Y. Wu, J.A. Stankovic, T. He, S. Lin, Realistic and efficient multi-channel communications in wireless sensor networks, in *INFOCOM* (2008)
5. K.L. Hieu, H. Dan, A. Tarek, A practical multi-channel media access control protocol for wireless sensor networks, in *ACM/IEEE International Conference on Information Processing in Sensor Networks* (2008)
6. H.W. So, G. Nguyen, J. Walrand, Practical synchronization techniques for multi-channel MAC, in *MobiCom* (2006)
7. L. Kong, X. Liu, mZig: enabling multi-packet reception in ZigBee, in *ACM MobiCom* (2015)
8. R. Ahmad, O. Sidek, S.K.K. Mohd, Development of the OQPSK modulator for ZigBee standard on FPGA, in *Proceeding of International Conference on Robotics, Vision, Signal Processing & Power Applications (RoViSP'09)* (2009)
9. C.M. Liang, N.B. Priyantha, J. Liu, A. Terzis, Surviving Wi-Fi interference in low power ZigBee networks, in *ACM SenSys* (2010)
10. A. Woo, D. Culler, A transmission control scheme for media access in sensor networks, in *MobiCom* (2001)
11. IEEE Computer Society, Local and metropolitan area networks—specific requirements Part 11: wireless LAN Medium access control (MAC) and physical layer (PHY) specifications. Available at: http://standards.ieee.org/getieee802/download/802.11-2007.pdf
12. IEEE Computer Society, Local and metropolitan area networks—specific requirements Part 15.1: wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). Available at: http://standards.ieee.org/getieee802/download/802.15.1-2005_part1.pdf
13. H. Rahul, N. Kushman, D. Katabi, C. Sodini, F. Edalat, Learning to share: narrowband-friendly wideband networks, in *ACM Sigcomm* (2008)

Chapter 3
Interference Model and Measurement

This chapter focuses on experiments and models for the signal interference. We will introduce the method for measuring cross-technology interference from WiFi on ZigBee nodes and present current models.

From the introduction in Chap. 2 we can see that WiFi and ZigBee have different communication technology and thus they cannot decode the signal from each other. Therefore, the first question we are facing is how to describe and measure the cross-technology interference.

3.1 Interference Measurement Methodology

In order to find the effect of cross-technology interference, the measurement methodology needs to be carefully designed. With different observation aims, several experiment methodologies have been used in past research. Due to the inaccuracy of existing simulators, most experiments are carried out using real network devices. This section makes a summary for these methods.

3.1.1 Interference Source

These measurement methods can be classified into two categories based on the interference source, i.e., controllable interference source and existed interference source. When controllable interference sources are used, the measurements are conducted under two networks: a *primary network*, which is always a Zigbee network and a *competing network* using WiFi with different working channel. Generally the distance, traffic amount and working frequency of the competing network can be easily set for observing different interference results. In the experiment, each network consists of two nodes communicating with each other so

G. Shi and K. Li, *Signal Interference in WiFi and ZigBee Networks*,
Wireless Networks, DOI 10.1007/978-3-319-47806-7_3

that the interfered traffic could be collected to be analyzed. In the case of WiFi, one of the two network nodes is an 802.11 AP and another is client.

The aim of controllable interfering experiment is to search the deep mechanism beyond the external phenomenon. But in reality, the interference traffic cannot be controlled and have some randomness. The final anti-jamming protocol or system must take the features of real traffic into account. Therefore, the interference measurement should also be conducted under existed interference source, such as enterprise WiFi network. Since these networks are not easy to control, traffic traces are often used.

3.1.2 Experiment Topology

For simplicity, most experiment measurements are conducted using single hop networks. This topology setting can directly measure packet loss caused solely by interference rather than artifacts caused by network protocols associated with multi-hop network paths (i.e., routing and MAC protocols) [1]. A typical network topology for WiFi interference measurement is illustrated in Fig. 3.1 where d_1 and d_2 vary to measure different interference level. Generally, this network topology is used to measure the interfered performance under a controlled interference. Thus, external interference will be removed in experiments.

Sometimes, we want to measure the interference condition for multi-hop ZigBee network to examine the interference variety with different locations. In this case, one multi-hop ZigBee network is needed to be setup and several WiFi communication links are needed to be deployed along the ZigBee network. A possible network setup for multi-hop interference measurement is illustrated in Fig. 3.2, where d_1 and d_2 can be changed as required.

3.1.3 Interference Traffic Control

For extensive experiment, different interference condition should be examined by adjusting the network parameters. Generally, the following configurations for interference measurement are frequently used in experiment [2].

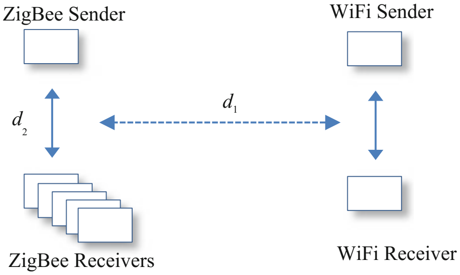


Fig. 3.1 Typical setup for one-hop interference measurement

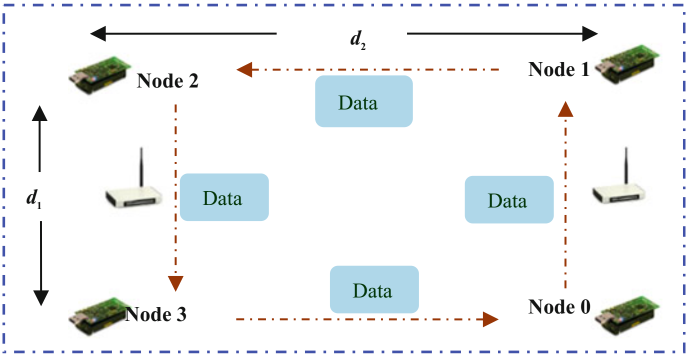


Fig. 3.2 Setup for multi-hop interference measurement

- *Working frequency*

For analyzing the interference lever, we expect that transmissions occurring in different WiFi channels will cause varying levels of interference on the primary 802.15.4 channel. At the same time, we expect that transmissions occurring at different 802.15.4 channels will not likely interfere given the band-gaps between consecutive channels. Therefore, the channel of the primary network should be adjusted to examine the interference varying law by setting working frequency distance. Of course, the competing channels are also able to be changed depending on the interference setting.

- *Transmission power*

As mentioned in Chap. 2, the maximum transmission power of WiFi networks is 100 times larger than that of 802.15.4 networks and is thus likely to inundate the primary receiver. In reality, the interference level varies with distance from the interfering source. Thus, to evaluate at which point the primary receiver's radio can reliably decode the packet, the transmission power of the competing transmitter is needed to be changed as required.

- *Transmission rate*

Intuitively, the higher the rate of competing traffic, the larger the probability that the interference level in the primary receiver will be high is, thus causing more packet losses. Adjusting the sending rate of the competing network flow is also a method to make the primary network be exposed to different interference level. In all cases, the competing WiFi sender uses maximum MTU packets and transmits at the maximum link rate (e.g., 54 Mbps for 802.11g). By setting different transmitting rate of competing network in experiment, it is possible to examine the transmission ability of ZigBee network when facing interference.

- *CSMA/CA On/Off*

Since in wireless application only one device is allowed to transmit data at the same time, all devices would have to contend for the medium using a CSMA/CA mechanism, where a device wishing to transmit data must initially sense the radio medium to determine whether the channel is available. To minimize the probability of collision with other transmitting devices, all wireless nodes have to spend time on listening, waiting, and back-off, which impedes the data transmission and decreases the channel occupation rate. Thus, it is impossible to maximize the throughput of competing traffic if CSMA is disabled in experiment. At the same time, disabling CSMA of ZigBee can illustrate the impact of WiFi interference to ZigBee transmissions across the spectrum, while enabling CSMA can demonstrate the coordination effect of CSMA under WiFi interference. Therefore, enabling/disabling CSMA/CA is an important tool for interference measurement.

3.1.4 Experiment Device

Common devices used in interference measurement include types: COTS ZigBee or WiFi devices and Software defined Radio (SDR) devices [3]. COTS devices are easy to control but have some limitations due to the system implementation. Most COTS devices can be only accessed beyond PHY layer. In the last years, many researchers have also proposed the use of software-defined radio devices, such as USRP and WARP. SDR devices have the benefits of easy reconfiguration and additivity. For example, the information that cannot be obtained in COTS devices is possible to be collected in USRP devices. The choosing of experiment devices is depending on the requirement of experiment.

3.2 Interference Representation

Interference will bring performance impact with different level. When conducting experiments, we should pay attention to some parmeters for interference representation. In the following text, we will introduce these characteristic representation.

3.2.1 RSSI

Received Signal Strength Indicator (RSSI) can be obtained by reading a specific register for most wireless communication chip, such as theros [4] for WiFi or CC2420 [5] for ZigBee. It is closely related with the signal energy sensing on the receiver. Generally, the larger RSSI value, the higher the received signal energy is. But RSSI includes the energy of interference, the desirable signals, and the noise. If there are no traffic in primary network, RSSI can represent the level of interference.

3.2.2 Network Throughput

Interference has impact on network throughput significantly due to frequent collision, waiting, and channel sensing. The most direct result of interference is the decreasing of network throughput. Sometimes, we measure the throughput of a transmission link as the ratio between the receiving data rate (i.e., the number of bits received in one time unit) at the receiver and the nominal data rate at the transmitter. For example, the nominal data rate of MicaZ mote is 250 kbps, which is the maximum speed at 2.4 GHz for ZigBee.

3.2.3 Packet Reception Ratio

The standard definition of Packet Reception Ratio (PRR) is the radio between the number of received data packets and the number of transmitted data packets. Interference causes packet loss, which triggers the retransmission. Thus PRR also directly reflects the inference level.

3.2.4 Bit Error Pattern

Bit error patterns are helpful to understand what happened to the transmission. Recognizing these patterns will enable us to identify the channel condition in great details and can potentially bring benefit to channel coding, routing, and error correction protocol design. Most of the above representations have focused on high-level metrics when examining the interaction between 802.11 and 802.15.4 networks. Instead of limiting ourselves to observing only corrupted packets or packet error rates, bit error pattern provides more in-depth observation on how transmitted symbols are transformed upon corruption.

3.3 Theoretical Model

This section introduces the well-known theoretical model for characterizing signal interference.

3.3.1 SINR

Signal-to-interference-plus-noise ratio (SINR) is generally considered to be the key factors that indicate the quality of a wireless link. However, RSSI inaccurately captures the link quality and it is difficult to accurately compute SINR with commodity wireless cards. This result is easy to understand because RSSI reflects the energy level of signal at the nodes without distinguishing interference signal from the received mixed signals.

The standard SINR equation for each bit of a packet x that the receiver receives at time t is

$$\mathrm{SINR}(x,t) = \frac{S(x,t)}{I(x,t) + N_{\mathrm{env}}} \tag{3.1}$$

Interference $I(\cdot)$ is sum of all undesirable signals $S(y, t)$ (both external interferers and self-interference due to multipath) that arrive at the receiver at time t:

$$I(x,t) = \sum_{y \neq x} S(y,t) \tag{3.2}$$

We can ignore multipath in our line-of-sight setup, so $I(\cdot)$ is simply the instantaneous interferer power. The noise term in Eq. 3.1 has several components, but is mainly the channel and antenna noise. It is Gaussian in nature, and can be approximated as $N_{\mathrm{env}} = kTB$, where k is the Boltzmann constant, T is the receiver temperature, and B is the signal bandwidth. At room temperature, for 22 MHz 802.11b or 20 MHz 802.11g, N_{env} is about −100 dBm. For the 1 Mbps rate (the slowest possible), we can then calculate using standard formulas that we need a signal-to-interference ratio of at least 10 dB above this noise threshold of −100 dBm in order to achieve a Bit Error Ratio (BER) of 10^{-6} (which roughly corresponds to a 1 % packet loss with 1000 byte packets) [6].

3.3.2 BER-PRR Model

In ZigBee specification, the PHY at 2.4 GHz uses offset quadrature phase shift keying (OQPSK) as the modulation model. Denote that the E_b/N_0 is the ratio of average energy per information bit to the noise power spectral density at the receiver input, in the case of an additive white Gaussian noise (AWGN) channel. According to [7], the BER, can be calculated by the following equation:

$$\mathrm{BER} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{3.3}$$

where E_b/N_0 is the normalized signal-to-noise ratio (SNR) and $Q(\cdot)$ is Q-function of Gaussian distribution

$$Q = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} \exp\left(-\frac{u^2}{2}\right) \mathrm{d}u \tag{3.4}$$

When a ZigBee channel overlaps with a WiFi channel, we can consider the WiFi signal as partial band jamming noise for the ZigBee signal [8] and the SNR is replaced by SINR which can be defined as the following according to Eq. 3.1.

$$\mathrm{SINR} = \frac{P_{\mathrm{Signal}}}{P_{\mathrm{noise}} + P_{\mathrm{interference}}} \tag{3.5}$$

where P_{Signal} is the power of the desired signal at ZigBee receiver, P_{noise} is the noise power, and $P_{\mathrm{interference}}$ is the received interference power from WiFi signal at ZigBee receiver. Considering that the power spectrum of IEEE 802.11b is 11 times wider than ZigBee and is not uniformly distributed, in-band interference power of IEEE 802.11 cannot be simply calculated by dividing 11. An amendment parameter of in-band power factor r is added to $P_{\mathrm{interference}}$ [9]. Therefore, Eq. 3.4 is modified as

$$\mathrm{SINR} = \frac{P_{\mathrm{Signal}}}{P_{\mathrm{noise}} + r \cdot P_{\mathrm{interference}}} \tag{3.6}$$

In essence, any performance prediction model based on SINR is taking bit error rate (BER) as a starting point to get the desired performance metric via a series of complicated calculations [10]. For a fixed modulation and coding scheme and with interference treated as noise, a well-accepted model for packetized transmissions is that they succeed if the SINR exceeds a certain threshold β. Therefore, the probability that a transceiver successfully receives an incoming bit, denoted by $p(\tau)$, is governed by the following model:

$$p(\tau) = \mathrm{Prob}\left[\frac{P_{\mathrm{Signal}}}{P_{\mathrm{noise}} + r \cdot P_{\mathrm{interference}}} > \beta\right] \tag{3.7}$$

where β is a constant determined by the modulation scheme and the transceiver sensitivity. Unfortunately, the above BER-SINR model cannot be directly measured on commodity radio transceivers.

3.3.3 PRR-SINR Model

Currently, the PRR-SINR model is usually the most commonly used to assess packet reception rate of ZigBee link under the existence of WiFi sources. This model is a measurement-based packet-level interference model adopted from BER-SINR model. It correlates PRR instead of BER with SINR. In this model, the probability that a transceiver r successfully receives an incoming packet ω is given by

$$p(\omega) = f\left(\frac{\mathrm{RSS}(\omega)}{\mathrm{RSS}(\mathrm{Interference}) + n_{\mathrm{aver}}}\right) \tag{3.8}$$

where function $f(\cdot)$ can be determined by the measurements of SINR and PRRs. n_{aver} is the measured *average* power of ambient noise.

After adaption on the BER-SINR model, the PRR-SINR model can be easily measured on most commodity radio transceivers. What's more, the RSS values and n_{aver} can be obtained from a radio hardware register called RSSI that is available on commodity wireless platforms. Thus, (ω) can be measured as the practical link-level PRR. Moreover, the PRR-SINR model is critical for optimizing wireless protocol performance because it can *predict* the PRR of a link when it experiences interference.

3.4 Interference Measurement Result

This section presents some measurement results in previous publication. We will introduce the results from two aspects: the interference results of WiFi on ZigBee and ZigBee on WiFi.

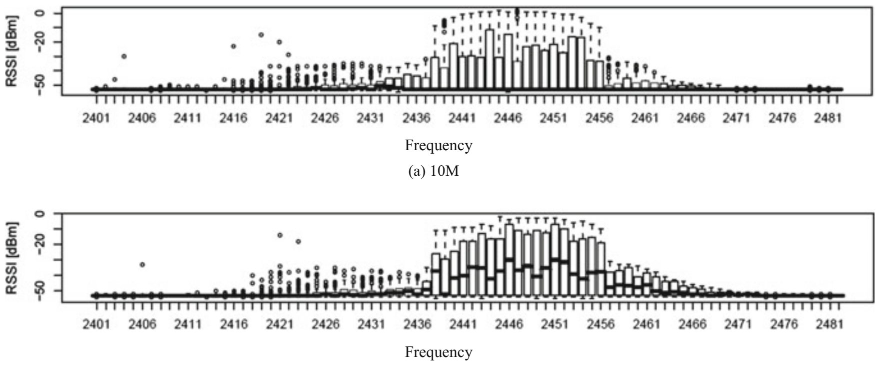
3.4.1 WiFi Interference on ZigBee

3.4.1.1 RSSI Variety on ZigBee

Studying the variety of RSSI on interfered ZigBee is the most common method. Figure 3.3 illustrate the impact of the WiFi traffic on the following RSSI metrics: average, Inter-Quartile Range (IQR) and outlying RSSI values [2]. The horizontal lines inside the IQR are the average values which are mostly visible in the graphs corresponding to the higher data rates. The IQR, which is the range covered by values lying between the first (25 %) and third (75 %) quartiles, is depicted by a vertical box for each frequency measured. Finally, the outliers are defined as measurements that are more than 1.5 × IQR away from the closest quartile and are depicted using small circles in Fig. 3.3. The *y*-axis corresponds to the RSSI_VAL raw dBm readings provided by the CC2420 radio (the transform from the original reading value to dBm has been introduced in Chap. 2).

During the measuring process, no transmissions occur in the primary network and there exist two WiFi networks with channel 8 and channel 4, respectively. From the figure we can see that there are two frequency ranges exhibit higher RSSI values. The rightmost range coincides with the frequency range of the 8th WiFi channel and the other corresponds to WiFi channel 4. It is also evident that while maximum recorded RSSI values increase across all data rates, the 10 Mbps transfer causes a significant change in IQR, while the 20 Mbps competing transfer increases the average RSSI value significantly.

This experiment results show us intuitively the variety of RSSI or signal energy on ZigBee nodes exposed to WiFi interference. It indicates that the RSS is not a reliable metrics to infer the link quality.



(a) 10M

(b) 20M

Fig. 3.3 Measured RSSI on the ZigBee network with varying WiFi sending rate on the competing WiFi network [2]

3.4.1.2 Packet Loss Rate

With the same experiment configurations, Fig. 3.4 illustrates the packet loss when transmission rate on the competing WiFi channel varies. The loss rate clearly reveals that these Zigbee channels overlapping with WiFi channel 8 are affected, and the center ones (18–21) show losses higher than 20 % in the worst case.

Using the experiment topology as illustrated in Fig. 3.1 where $d_1 = 12$ ft and d_2 varies from 15 to 170 ft, the percentage of different packet status is illustrated in Fig. 3.5. Different from the results in Fig. 3.3, this figure shows three types of packet reception events on ZigBee nodes: packets that are received correctly, packets that fail the CRC tests due to corrupted bits, and packets that are lost (i.e., transmitted but never received). As expected, ZigBee PRR is significantly reduced due to WiFi interference, especially when the two networks are closer to each other. As d_2 increases, the WiFi PRR improves since the interference becomes progressively weaker. This result presents the deep understanding of PRR and sheds light on the packet loss reasons.

3.4.1.3 Bit Error Patterns

In transmission of ZigBee PHY layer, one bit of data will be finally converted to a codeword (PN-Code) with 32-bit length. There are various methods to find out a 'best-match' PN-Code to decode received chip sequences to corresponding information bits. The maximum likelihood decoder (MLD) is the one. The detailed has been introduced in Chap. 2. Denote *chip errors per PN-Code* as CEPP. If putting all CEPPs of a frame together, a time series of CEPPs can be obtained which is denoted by Λ. The ith CEPP in the time series is denoted by $\Lambda(i)$.

Figure 3.6 depicts several chip error patterns [12]. The x-axis is the index of CEPP (from 1 to 146) and the y-axis is the value of CEPP. A red bar means the

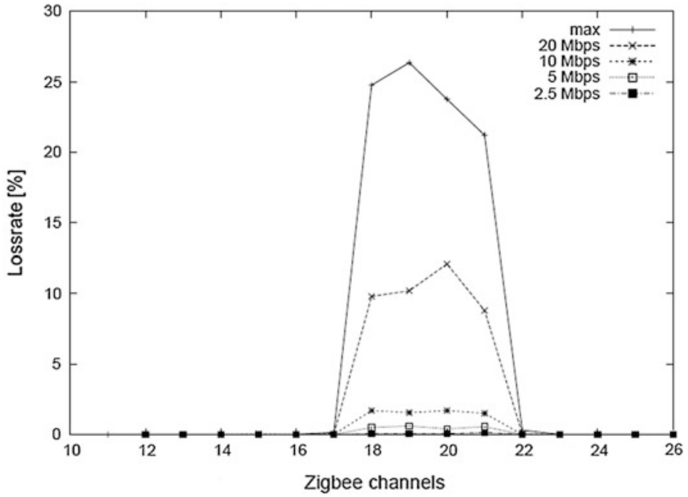


Fig. 3.4 The packet loss varies with transmission rate of interference [2]

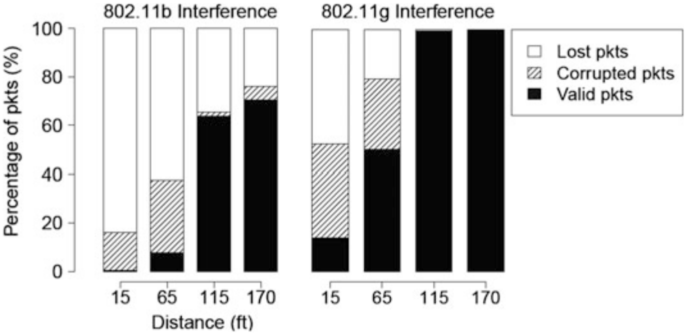
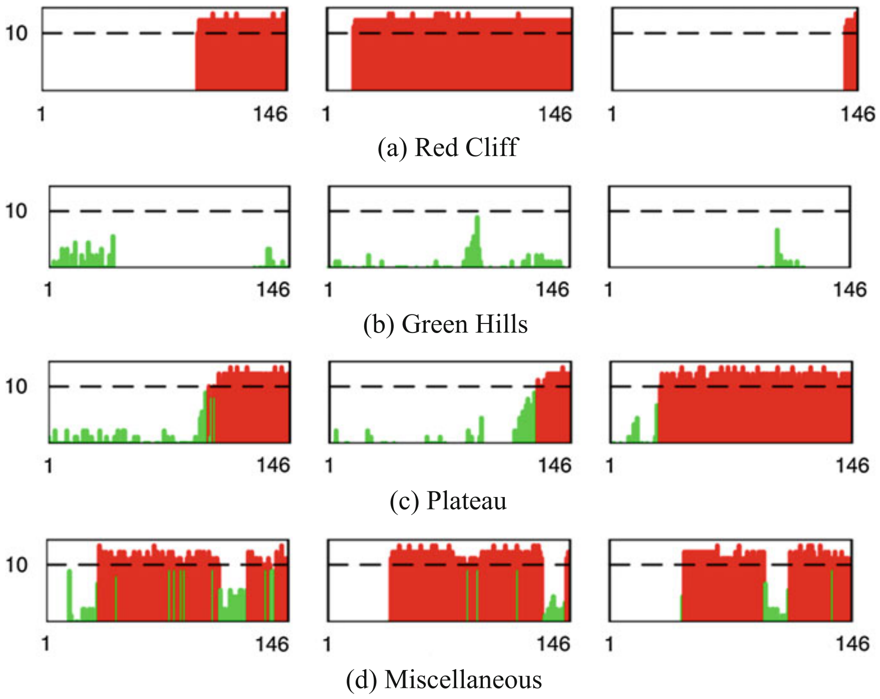


Fig. 3.5 Percentage of ZigBee packets correctly received, corrupted, and lost as the distance between WiFi nodes and ZigBee nodes increases [11]

corresponding CEPP is greater than or equal to the threshold while a green bar means the contrary. Each graphic is tagged with its pattern name.

3.4.1.4 Characteristic of WiFi Traffic

Many research results have shown that WiFi traffic is highly bursty as t wide range of time scales. Figure 3.7 shows the arrival of WiFi frames are clustered together with short intervals. This means that the network traffic is highly bursty and will leave significant amount of white spaces between 802.11 frames.



(a) Red Cliff

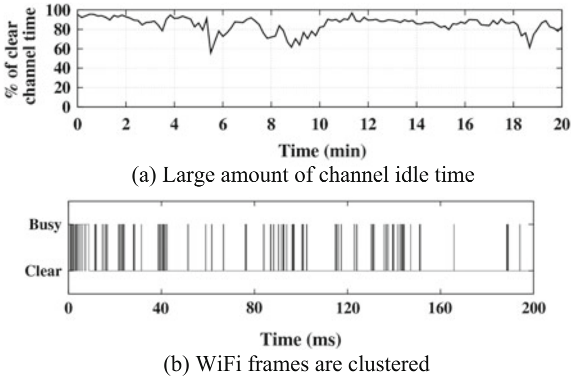
(b) Green Hills

(c) Plateau

(d) Miscellaneous

Fig. 3.6 Four major chip error patterns [12]

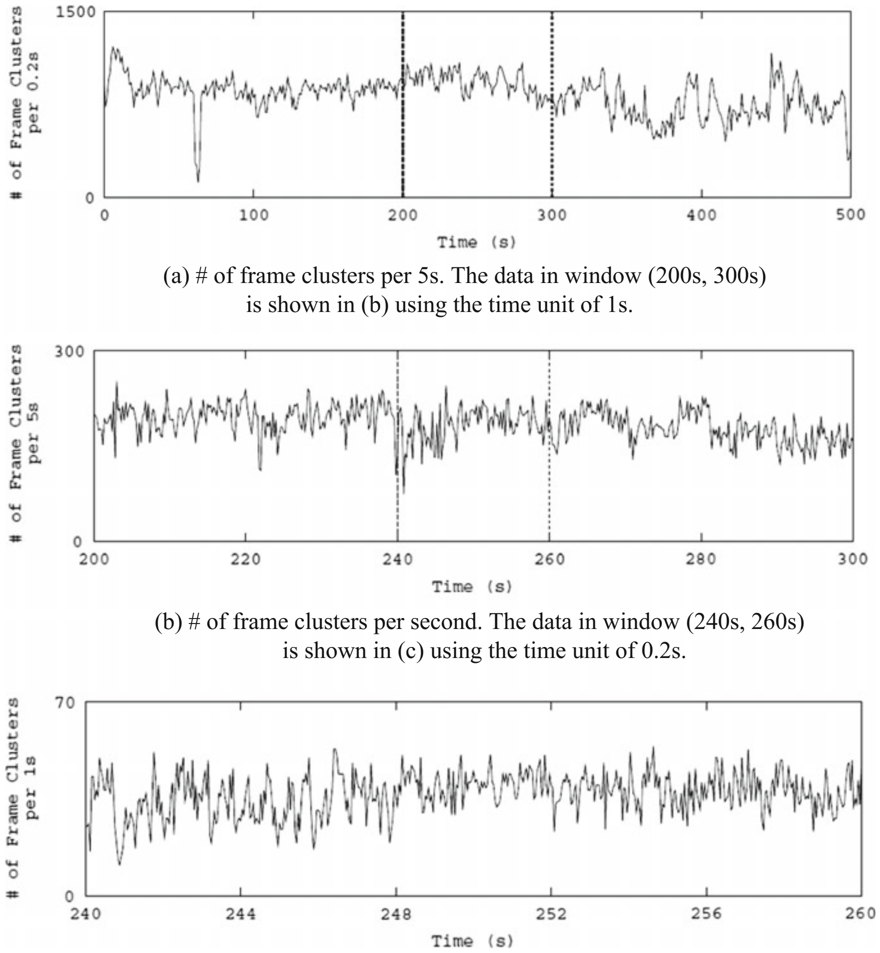
Fig. 3.7 WiFi bursty traffic in real scenarios [13]



(a) Large amount of channel idle time

(b) WiFi frames are clustered

Figure 3.8 plots the scaling behavior of the frame cluster arrival process [13]. The number of arrived frame clusters for three time units: 5, 1, and 0.2 s. The plots show similar variance at all time scales. This time-scale invariant feature suggests that the arrival process of WiFi frame clusters is self-similar. Self-similarity means that “traffic looks the same on all time scales” [14].



(a) # of frame clusters per 5s. The data in window (200s, 300s) is shown in (b) using the time unit of 1s.

(b) # of frame clusters per second. The data in window (240s, 260s) is shown in (c) using the time unit of 0.2s.

(c) # of frame clusters per 0.2s.

Fig. 3.8 Self-similarity of 802.11 frame cluster arrival process [13]

3.4.2 ZigBee Interference on WiFi

Traditional experiments and analysis regularly neglect the impact of ZigBee on WiFi networks due to the low output power. However, it is not the output power but the received power that governs interference, and this is a function of the output powers but also the communication distance. If the distance between WiFi and ZigBee is close to each other, the interference from ZigBee has also impact on WiFi performance. The following observations contradict the common belief that WiFi nodes do not back-off in the presence of ZigBee traffic.

In the literature [14], the impact of ZigBee on WiFi was studied based on the topology illustrated in Fig. 3.9. In experiment, ZigBee interferer is transmitting packets of 127 bytes in length with a channel occupation of 5 %. The results show that the interference impact on 802.11b can be seen only when the offset between the central frequencies is 2 MHz and the 802.11b packet length is longer than 600 bytes. However, interference on the IEEE 802.11g transmission is not noticeable. The authors infer that it is most likely due to the robustness of the OFDM modulation used.

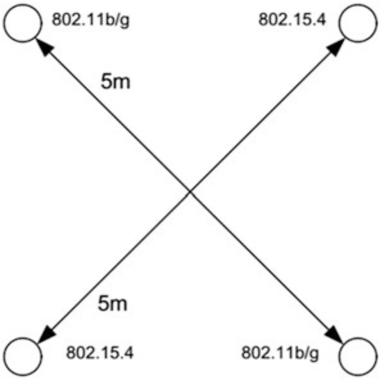
In literature [15], the packet loss rate and throughput for an 802.11b network when interfered with by 802.15.4 traffic are evaluated. The results show that when the distance between the 802.11 receiver and 802.15.4 transmitter is small, performance degradation can be large.

The literature [16] examines the impact of ZigBee interference on WiFi. To test this, the 802.15.4 transmitter was placed close to the 802.11 receiver. The respective noise powers for 802.11 packets and 802.15.4 packets as measured by the spectrum analyzer. Figure 3.10 shows the results when the 802.15.4 transmitter sends at 640 packet/s. The intervals with 802.15.4 traffic can be seen clearly from the picture. For each of the 802.11 physical layer bit rates, the actual throughput is lower whenever the 802.15.4 transmitter is on. This experiments show that the low-power ZigBee has also impact on the performance of WiFi, although the impact only happens when they are close.

In the experiment of Literature [11], the same phenomenon is also observed. The experiment is carried out using the topology illustrated in Fig. 3.1 where $d_1 = 15$ and $d_2 = 12$ ft. Figure 3.11 presents a timeline of 802.11b and ZigBee traffic activity. Each vertical box corresponds to a single ZigBee broadcast, while the grey region corresponds to 802.11b activity. One can see from Fig. 3.10 that 802.11 back off during ZigBee transmissions, when the separation between 802.11 and 802.15.4 nodes is small.

This behavior can be attributed to the 802.11 specification that mandates performing a clear channel assessment (CCA) prior to every data packet transmission.

Fig. 3.9 Experiment topology used in [14]



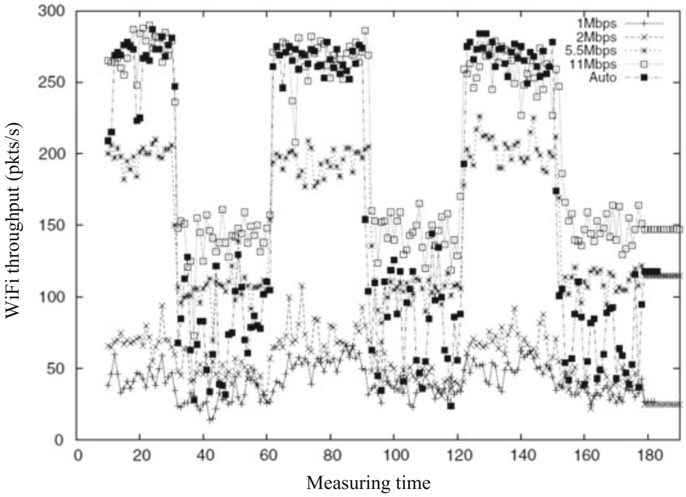


Fig. 3.10 WiFi throughput versus time when ZigBee rate set to 640 packet/s [16]

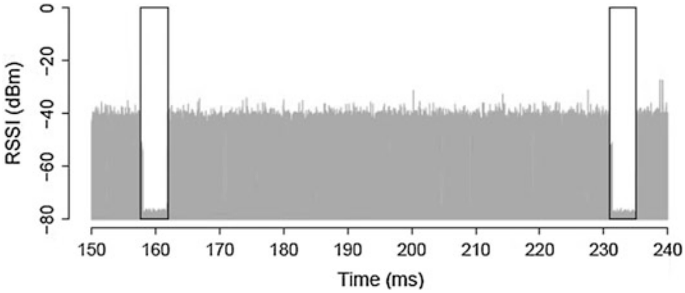


Fig. 3.11 Overlay of 802.11b and 15.4 traffic where the gray lines are RSSI measurements corresponding to 802.11 transmissions [11]

In other words, the 802.11b radio in the experiment sensed the channel noise floor being above the CCA threshold and deferred its pending transmission.

We also note that in certain cases, the 802.11 radio will not back-off. First, the 802.11 specification requires that RSSI $\gg -70$ dBm for the channel to be considered busy when TX power $\ll 50$ mW. Second, the 802.11 specification lists three CCA modes (i.e., energy detection, packet detection, and both) and vendors can implement one or more at their discretion. 802.11 radios that use the packet detection CCA mode will declare the channel to be clear, since they cannot decode the overheard 15.4 packet transmission.

We can now explain why 15.4 and 802.11 packets collide. Looking at Fig. 3.12, the only time that a 15.4 sender can begin its transmission is during the DIFS + ContentionWindow period since it otherwise senses the channel as busy.

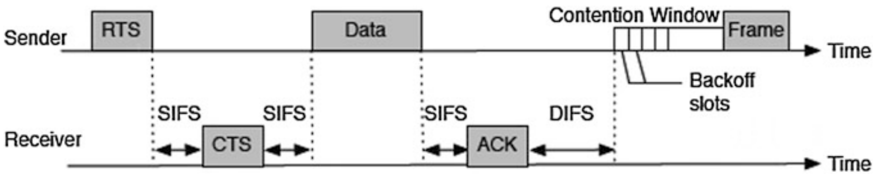


Fig. 3.12 Messages and delays defined in the 802.11MAC protocol. Durations of packet transmissions and time intervals depend on the 802.11 variant used. The leading RTS/CTS exchange is used only for large packets [11]

Furthermore, the time granularity that the 15.4 sender senses the medium is equal to the slot time (=320 μs) and it senses the medium for eight symbol periods (=128 μs) before declaring the channel as idle. Considering the short length of the DIFS interval and the shorter 802.11 slot time it is very likely that during the time the 15.4 sender senses the channel, the 802.11 node also senses the channel. As a result of both nodes sensing the channel idle, they start transmitting at the same time and subsequently collide.

3.5 Summary

This chapter discusses the problems of interference measurement and interference model. These results provide the foundation for the solutions in the following chapters. Unless there is a special requirement, we will refer to these results directly in the future discussion.

References

1. Z. Zhao, X. Wu, X. Zhang, J. Zhao, X. Li, ZigBee vs WiFi: understanding issues and measuring performances of their coexistence, in *IPCCC* (2014), pp. 1–8
2. R. Musaloiu-E, A. Terzis, Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. Int. J. Sens. Netw. **3**(1), 43–54 (2008)
3. R. Musaloiu-E, A. Terzis, Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. Int. J. Sens. Netw. (IJSNet) **3**(1), 43–54 (2007)
4. M. Ettus, *The Universal Software Radio Peripheral or USRP* (2008)
5. Atheros-datasheets, https://www.atheros-drivers.com/qualcomm-atheros-datasheets.html
6. CC2420—Texas Instruments, http://www.ti.com/lit/ds/symlink/cc2420.pdf
7. R. Gummadi, D. Wetherall, B. Greenstein, S. Seshan, Understanding and mitigating the impact of RF interference on 802.11 networks. SIGCOMM **37**(4), 385–396 (2007). B. Sklar, *Digital Communication* (Prentice Hall, 1995)
8. J. Mikulka, S. Hanus, Bluetooth and WiFi coexistence simulation, in *Proceedings of the 4th WSEAS International Conference on Information Security, Communications and Computers*, Tenerife, Spain, 16–18 Dec 2005

9. S.S. Wagh, A. More, P.R. Kharote, Performance evaluation of IEEE 802.15.4 protocol under coexistence of WiFi 802.11b. Proc. Comput. Sci. **57**, 745–751 (2015)
10. S. Liu, G. Xing, H. Zhang, J. Wang, J. Huang, M. Sha, L. Huang, Passive interference measurement in wireless sensor networks, in *Proceedings of the 18th IEEE International Conference on Network Protocols (ICNP'10),* 5–8 Oct 2010, pp. 52–61
11. C.-J.M. Liang, N.B. Priyantha, J. Liu, A. Terzis, Surviving wi-fi interference in low power ZigBee networks, in *SenSys* (2010), pp. 309–322
12. K. Wu, H. Tan, H.-L. Ngan, L.M. Ni, Chip error pattern analysis in IEEE 802.15.4, in *IEEE INFOCOM* (2010)
13. J. Huang, G. Xing, G. Zhou, R. Zhou, Beyond co-existence: exploiting WiFi white space for ZigBee performance assurance, in *ICNP* (2010), pp. 305–314
14. M. Petrova, J. Riihijarvi, P. MAhonen, S. Labella, Performance study of IEEE 802.15.4 using measurements and simulations, in *WCNC* (2006)
15. D.G. Yoon, S.Y. Shin, J.H. Park, H.S. Park, W.H. Kwon, Performance analysis of IEEE 802.11b under multiple IEEE 802.15.4 interferences, in the series *Lecture Notes in Computer Science* 2007, vol. 4517, pp. 213–222
16. S. Pollin, I.L. Tan, B. Hodge, C. Chun, A. Bahai, Harmful coexistence between 802.15.4 and 802.11: a measurement-based study, in *CrownCom* (2008), pp. 1–6

Chapter 4
Interference Avoidance in ZigBee Networks

Facing higher power interference, an intuitive strategy for ZigBee networks is to seek the opportunity in space, time, and frequency to avoid the interference. In this chapter, we will focus on the methods of interference avoidance in ZigBee networks.

4.1 Introduction of Interference Avoidance

Different with homogeneous networks, interference signal cannot be decoded by the heterogeneous interfered nodes due to different physical technique. Therefore, interference avoidance is a common strategy for mitigating cross-technology interference. The meaning of interference avoidance is that the interfered network adjusts its used resource such as channel to avoid straight collision with the high-power nodes.

The basic process of interference avoidance is illustrated in Fig. 4.1. Generally, the nodes work in normal state once the network is initiated. Then the ZigBee devices periodically or randomly detect the interference nearby and determine whether the interference exists or not. If yes, a new channel having a lower interference is discovered to reduce the interference.

For avoiding interference, there are at least two components in interference management protocols: interference detection and channel switching. Interference detection is designed for measuring the interference level and the existence of interference so as to determine whether to change the working frequency or not. Channel switching is responsible to choose more appropriate frequency and coordinate the nodes to keep the links usable.

G. Shi and K. Li, *Signal Interference in WiFi and ZigBee Networks*,
Wireless Networks, DOI 10.1007/978-3-319-47806-7_4

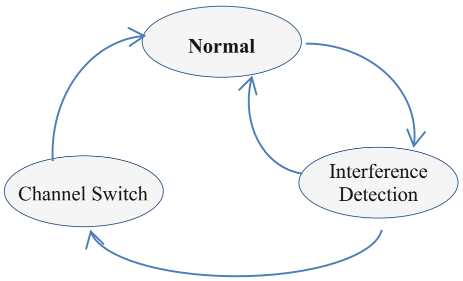


Fig. 4.1 Sate diagram for interference avoidance

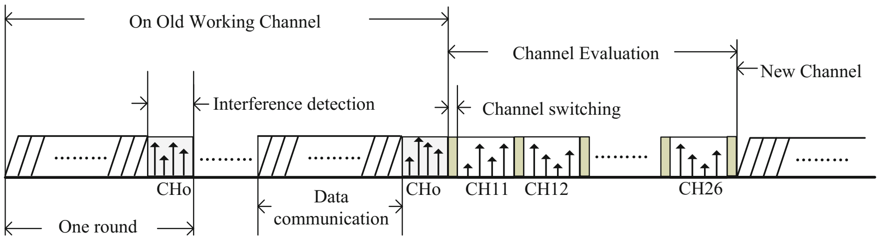


Fig. 4.2 Working procedure in ZigBee networks for interference avoidance

A possible working procedure for ZigBee networks with interference avoidance is illustrated in Fig. 4.2, where the segments with arrow denote the RSSI samples for interference assessment. Each node will periodically or nonperiodically assess the interference in its current working channel and on detecting interference searches a relatively clear channel as the new working channel.

4.2 WiFi Detection in ZigBee

Simply using RSSI could not differentiate WiFi interference from mix background signals, although all the signals in detected RSSI have interference on ZigBee. For example, the commercial microwave ovens also generate interfering radiation in the ISM band during their operation. In this section, we will discuss how to determine whether an interference signal is from WiFi.

Based on the introduction in Chap. 2, we have seen that IEEE 802.11 has some unique features, which could be used to figure out whether an interference source is from IEEE 802.15.4 node. These features include spectrum bandwidth, periodical beacon, signal power, and so on. These features have differences with other interference sources. In the following, we will introduce some method for WiFi detection.

4.2.1 Detecting WiFi via Power Spectrum

First, one can explore the statistics of RSS samples, such as power magnitude, time duration, and inter-arrival gap, to find distinctive features of WiFi traffic. However, these features vary significantly with environments, version of 802.11, and application traffic. Moreover, they may well resemble the interference from other RF sources such as ZigBee nodes or Bluetooth nodes transmitting on overlapping channels. Figure 4.4 shows the cumulative distribution of frame sizes and inter-arrival gaps of two WiFi and ZigBee traffic traces. The surprising resemblance between these two traces makes it challenging to reliably detect the existence of WiFi networks.

4.2.2 Detecting WiFi via Periodic Beacons

In Chap. 2 we have introduced the beacon information in WiFi networks. Periodic beacon broadcasting is mandatory in 802.11 infrastructure networks. The typical length of beacon frame ranges from 80 to 200 bytes depending on the amount of management information it carries. The periodic broadcasting could be used as WiFi signature to infer interference from WiFi.

As we have known that ZigBee nodes could not directly decode 802.11 frames, ZiFi [1], instead using coding information, searches for 802.11 beacon signals from RSS samples instead. To do this, RSS sampler running in ZigBee node reads the RSSI register of ZigBee radio every T *us* for total D *us*. T and D are referred to as RSS sampling period and sampling window size, respectively. The sampling period should be short enough to capture the transmission of 802.11 beacon frames. ZiFi is the first work to discover WiFi via sensing periodic beacon in ZigBee nodes.

After enough samples are collected, the RSS shaper screens the noise in RSS samples to improve the discovery accuracy. Instead remove these RSS noises from RSS sequences, ZiFi shapes all samples into 1, 0 sequence by setting the magnitude of these RSS samples to zero if it is below −90 dBm and setting that of all remaining RSS samples to 1. If there are S consecutive nonzero samples, this is typically generated by WiFi data traffic. In this case, these S consecutive nonzero samples will be set to zero too. After the above preprocessing, all the magnitude of RSS samples is either 0 or 1 finally. Suppose R represents the time series of N RSS samples and $R[i]$ ($i \in [1, N]$) is the RSS magnitude in the ith sampling instance. After the above processing, each item $R[i]$ in R is transformed into 1 or 0.

After shaping the raw RSS samples, ZiFi folds the 0, 1 sequence. For each folding operation, the sequences are added together in an element-wise fashion. For a RSS sequence R with a period of P, the folding process can be written as

$$F_p[i] = \sum_{j=0}^{\frac{N}{P}-1} R[i + j * P]$$

where $F_p[i]$ is referred to as the ith folding result and the maximum is referred to as the folding peak of period P. In this way, the magnitude of the sum will be amplified at a period of P while the sum of noise in the series is likely smaller due to their nonperiodicity if the phase of folding happens to align with that of the periodic signal. After N-P number of additions for each possible period, the maximum folding and P can then be found as the period that yields the maximum folding result.

For example, suppose the ZigBee node has collected the RSSI with 24 samples, as illustrated in Fig. 4.3 where series of RSSI samples expressed as boxes. Both black and gray indicate a busy channel where the former is a periodic beacon signal, while the latter is random noise induced by traffic or interference. Denote the number of samples in the interval of the beacons as λ and in the example λ is 6. Upon folding by $P = λ$, RSSIs of beacons (in black) align in a column. The column with the highest fold sum indicates the position of the beacon.

In order to handle the unknown period and their wide ranges, ZiFi proposed a novel folding algorithm named with **Common Multiple Folding**. Since we are focusing on signal interference we do not discuss the detailed algorithm of CMF. For details, please refer to literature [1]. Figure 4.4 shows the signals before and after applying CMF algorithm. Literature [2] further uses CMF for mobile devices location.

Detecting WiFi in ZigBee nodes provide a more accurate way to determine whether an interfering signal comes from WiFi or not. However, it is very difficult to do this in low-cost embedded devices. In fact, ZiFi requires ZigBee nodes to be connected with a computer to run the signal processing algorithm. As result, using this mode to detect WiFi signal is not efficient for networked ZigBee nodes.

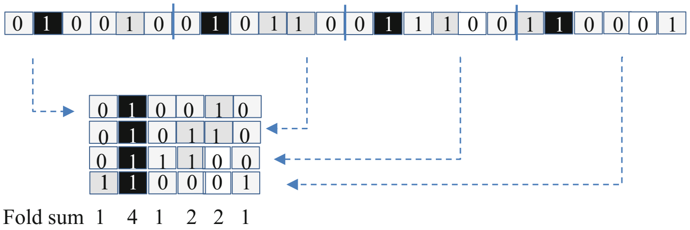
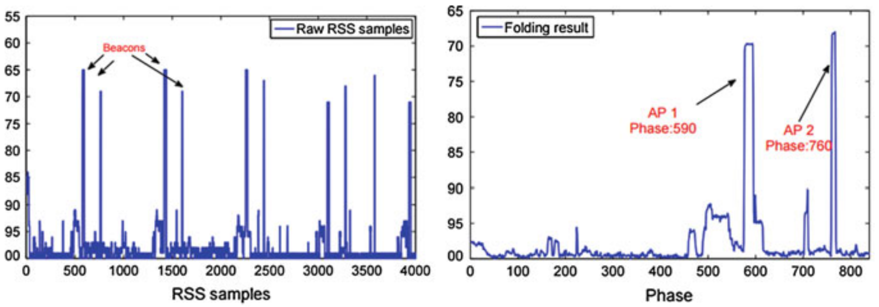


Fig. 4.3 Folding example



(a) Raw RSS samples collected in a noisy channel.

(b) The folding result of period 102.4ms.

Fig. 4.4 CMF discovers the beacons in noisy RSS samples and groups them based on identities [1]

4.2.3 Detecting WiFi via Corruption Patterns

Different interference sources have different interference features and each interferer has characteristic patterns which are embodied in Link Quality Indicator (LQI), signal strength, and corrupted parts in receipted packet [3, 4]. If we could make classifications based on these observed patterns, we might be able to make a distinction between serval interferers such as WiFi, Bluetooth, and microwave oven.

In a controlled experiment, Frederik Hermans et al. demonstrate a clear pattern of corrupted packets for different interferers [3]. However, it is difficult to classify the type of interference using only one feature in real environment. Rather, it is the combination of multiple features that facilitates classification. Following this idea, a number of features in the course of this work are used in classification via supervised learning approach.

More specifically, the features could be used in interference classification include:

- **LQI threshold**: LQI reflects the chip error rate over the first two bytes of a packet. If a packet is received with a high LQI, but the packet fails the CRC check, this indicates that channel conditions were good when reception started, but then deteriorated. In contrast, when a packet has a low LQI value, the channel conditions are poor over the whole packet reception time. When this happens, packets will not be decoded correctly due to insufficient signal strength. In [3] the threshold value 90 was chosen empirically from experiment measurement data.
- **RSSI-related features**: For packets that are corrupted due to insufficient signal strength, the RSSI values vary little, whereas distinct peaks in signal strength can usually be seen for interfered packets. In [3], a binary feature is defined to indicate whether the range of RSSI values is greater than 2 dB. Before apply this feature, the series of RSSI measurements is normalized by the maximum value in the series. This processing is allowed to abstract from the concrete distances and transmission powers while preserving the series' shape.

- **Number of corrupted symbols**: Radio technologies such as Bluetooth or 802.11g specify different bit rates, minimum and maximum packet lengths, as well as inter-packet delays for medium access control. These specifications put a constraint on the amount of damage that can be done to an 802.15.4 packet [3] found the distributions of WiFi and Bluetooth have distinct means, whereas the distributions for microwave interfered packets and packets with insufficient signal strength are more similar. This feature helps to distinguish different types of interferers.
- **Error bursts**: An error burst is a sequence of corrupted symbols. The MAC protocols of WiFi and Bluetooth impose constraints on the time between two packets. Thus, considering the spacing between two error bursts, i.e., the number of symbols correctly received between two error bursts, can give away useful information ab out the interferer. In [3], the experiments observe regular bursts of specific lengths for WiFi interference and Bluetooth interfered packets often contain two error bursts of specific length.

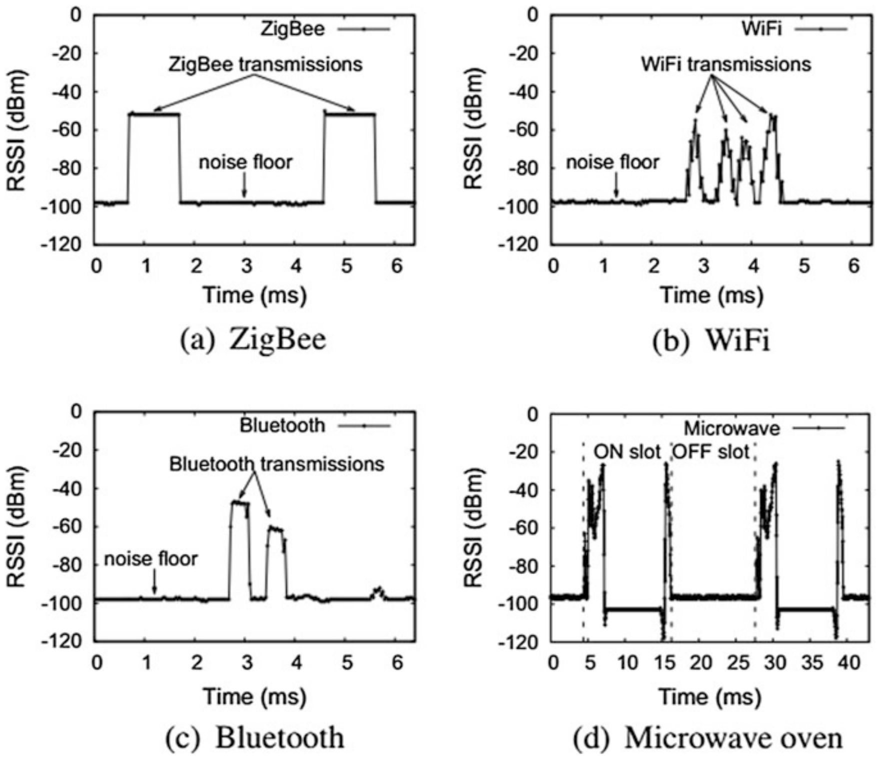
Combining the above features, the literature [3] further proposes to use a supervised learning approach to train a classifier to assign each corrupted packet to a class representing WiFi, Bluetooth, or microwave interference, or corruption due to insufficient signal strength. The detailed classification rules are listed in the Table 4.1.

In supervised learning, a classifier is trained on a set of examples for which the correct class is given. However, the learning phase is computationally more costly than classifying individual packets. Therefore, this method could not work in a real-time way in low-cost sensor.

4.2.4 Detecting WiFi via RSSI Pattern

There indeed exist several features that can be leveraged to effectively identify data transmissions for different radio technology. These demonstrated features are exposed by network standards, hardware specifications, modulation methods, etc. For example, the on-air time of a normal ZigBee data packet in TinyOS for CC2420 radio chip is between [576 μs, 4256 μs]. In contrast with ZigBee, WiFi, and Bluetooth have a shorter on-air time, while microwave ovens have a longer on-air time. At the same time, different protocol standards have different minimum packet interval due to the media access control. These factors lead to different RSSI pattern.

Literature [12] examines different RSSI patterns under the controlled environments to obtain their accurate characteristics. The experiments are done on CC2420 with 32us/sample. The results are shown in Fig. 4.5, where noise floor is the received signal strength of the background noise when there is no wireless activity in the channel. From this figure, we can see that ZigBee shows flat high RSSI



(a) ZigBee

(b) WiFi

(c) Bluetooth

(d) Microwave oven

Fig. 4.5 RSSI patterns of different 2.4 GHz technologies

segments and spikes are observed during one packet's transmission. This can be explained by their different physical modulation technique. The results show that the RSSI sequences of different wireless technologies exhibit different patterns.

Furthermore, in order to distinguish those patterns based on only RSSI information within a short time, literature [12] leverages the following features to identify ZigBee transmissions without incurring additional sampling overhead. These features include signal on-air time, minimum packet interval (MPI), peak-to-average power ratio (PAPR) and under noise floor (UNF). Table 4.2 lists these features. To extract the feature values in this table, consecutive samples with RSSI readings different from the noise floor are input to a carefully designed identification algorithm to identify the ZigBee segments. The basic deterministic algorithm takes feature vector **f** = ($PAPR$, T_{on}, MPI, UNF) as input to compare the exacted value with the rule value. To determine whether a segment is ZigBee or not, the algorithm has four conditions to be checked: (1) $C1$: $PAPR \leq PAPR_{\mathrm{ZigBee}}$; (2) $C2$: $T_{\mathrm{on}} \geq T_{\mathrm{min}}$; (3) $C3$: $|MPI - MPI_{\mathrm{valid}}| \leq \delta$; (4) $C4$: $UNF = FALSE$. and finally output the result. Take Fig. 4.6 as an example, the algorithm process is illustrated as follows:

Table 4.1 Three features for interference classification [3]

LQI-based	LQI < 90	A binary feature that indicates whether the LQI of a corrupted packet is higher than 90
RSSI series-based	Max (RSSI)—min (RSSI) > 2 dBm	A binary feature to indicate whether the range of RSSI values is greater than 2 dB
	Avg (RSSInormed)	Mean normalized RSSI
	Stddev (RSSInormed)	Standard deviation of normalized RSSI
	Max (RSSInormed)—mode (RSSInormed)	Difference between the maximum normalized RSSI value and the most common RSSI value
Error burst-based	Number of corrupted symbols/payload length	A sequence of corrupted symbols
	Avg (burst lengths)	Mean burst length for each packet
	Stddev (burst lengths)	Standard deviation of burst length
	End of last burst—start of first burst	Counts the number of symbols between the first burst's start and the last burst's end
	Avg (burst spacings)	Burst spacing mean to capture these variations

Table 4.2 Features for different wireless technologies

Wireless technology	On-air time	MPI	PAPR	UNF
ZigBee	[576, 4256] us	2.8 ms or 192 us	≤1.3	FALSE
WiFi	[194, 542] us	≥28 us	≥1.9	FALSE
Bluetooth	366 us	NA	≤1.3	FALSE
MWO	10 ms	10 ms	≥2.9	TRUE

(a) A flat segment with RSSI around the noise floor. The algorithm decides there is no ZigBee.

(b) A flat segment with RSSI above the noise floor. The algorithm decides ZigBee exists due to the valid condition vector (T, T, T, T).

(c) Two flat segments with RSSI above the noise floor. Even PAPR is small, the on-air time is too short to be ZigBee.

(d) Two fluctuant segments with RSSI above the noise floor. C1 and C3 are violated. Hence, the algorithm decides there is no ZigBee signal.

(e) A fluctuant segment with RSSI both above and below the noise floor. The algorithm decides it is not ZigBee signal as C1 and C4 are violated.

Although this method aims to detect ZigBee transmission, it is possible to recognize WiFi signal following the rules list in the Table 4.2. However, there is no related work to move on this way.

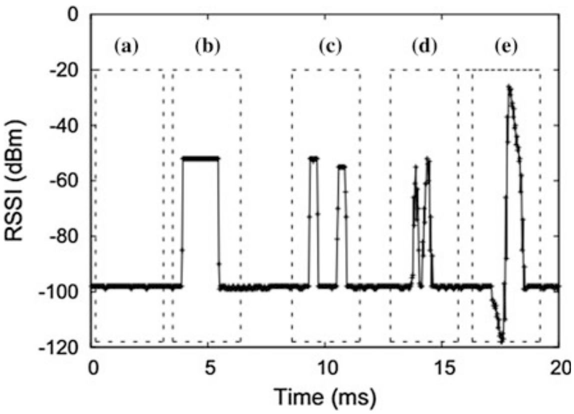


Fig. 4.6 The segment of RSSI, the dashed rectangles are sampling windows

4.3 Interference Detection

In Sect. 4.2, we have introduced several methods for WiFi detection. However, these schemes have little application in interference avoidance. This reason can be explained by the fact that the common interference sources in 2.4 GHz all have effects on ZigBee despite the type of inference type. Thus, it is unnecessary to accurately distinguish the interference sources for avoidance unless the solutions have targeted objects. It is enough if we can determine that the interference exists exactly. Therefore, interference detection is more important than interference classification.

In order to adjust working channel dynamically, it is necessary for ZigBee devices to have the ability to detect interference around them. Since ZigBee node cannot decode the WiFi signal directly, current interference detection commonly depends on signal energy level by using CCA mechanism (For CCA implementation, please refer to Chap. 2). By reading a register, RSSI information can be obtained. Ideally, in no interference environment, almost all RSSI readings are below a specific value H. Therefore, we can assume that all RSSI readings higher than H were from other wireless networks.

However, just like the surround noise, interference energy levels vary with time and once energy reading could only get current interference feedback. But ZigBee networks want to know the future interference condition for a better channel. As result, how to accurately predict the interference based on the readings from CCA is an important problem.

Before we could assert whether the interference exits, it is necessary to assess the interference level. Here we introduce a comprehensive method for interference assessment and then present an interference detection method.

4.3.1 Interference Assessment

Intuitively, signal interference involves not only interference intensity but also interference density [5]. In fact, SINR is only an intensity indicator and the conflict time ratio is a density indicator. Only the combination of these two aspects can have the severity of interference be assessed accurately. In addition to considering the characteristics of resource-constrained that embedded devices have, the interference assessment metric should respond promptly to environment change and be involved with less calculation and overhead.

Based on above introduction, a two-tuple <*u, v*> could be chosen to represent the intensity and density of interference, Here, *u* is a density indicator and *v* is an intensity indicator. Thus, we have a metric indicating the interference intensity and interference density simultaneously. Given the <*u*, *v*> pair, we can distinguish between different interference level by comparing *u* first and then *v* if *u* ties. We choose interference density as the primary key because that interference signal rarely occupying the channel has little effect on the original link's performance, even if it is very fierce.

In order to calculate the indicator, RSSIs series whose values are greater than *H* are chosen to be as the basis. Algorithm 1 [5] describes the detailed method for determining whether the interference exists or not. This algorithm samples RSSI reading on its current channel periodically and collects *W* RSSI readings in each round (Line 3). Based on the RSSI set, the number of RSSI samples that are bigger than *H* can be calculated out as well as the average value of these RSSIs (also beyond *H*) (Line 5). And then the final indicator pair <*u, v*> is taken as <*N/W*, *A/N*> (Line 7), where *N/W* is used to approximate channel occupancy rate and *A/N* as the intensity indicator.

Algorithm 1. Calculate_<*u*,*v*>

```
Input: Channel $i$;
Output: Interference pairs $<u,v>$
1 $S \leftarrow \Phi$, $N \leftarrow 0$, $A \leftarrow 0$;
2 repeat
3     sample RSSI reading $s_i$ on channel $i$;
4     $S \leftarrow \{s_i\}$;
5     if $s_i > H$, then $N{+}{+}$, $A \leftarrow A + s_i$;
6 until $|S| == W$
7 $u \leftarrow N/W$, $v \leftarrow A/N$;
```

The above method meets the required features for lightweight interference detection in low-cost sensor networks. The only overhead is to read RSSI register, which is a very cheap operation (about 0.37 ms per reading). It combines the density and intensity information to assess the severity of interference. More importantly, the method does not differentiate the interference traffic and takes all interference sources as uniform signal strength and thus is efficient especially when multiplex complex interference sources exist

4.3.2 *Interference Detection*

Using algorithm 1, we can readily detect the interference and assess the severity. However, the <*u*, *v*> pair obtained from Algorithm 1 only represents the interference information for one round. However, the interference might vary over time. In order to reflect the interference situation truthfully and respond to the variation quickly, we employ exponentially weighted moving average (EWMA) technology, which is used by wired network to estimate round-trip time. In this way, *u* and *v* is processed according to the following rule when algorithm 1 finishes:

$$X = (1 - \alpha) * X + \alpha * X' \tag{4.1}$$

Here, X' is the value for current round and X is the EWMA value. $\alpha = 0.125$ could be set empirically.

Algorithm 2. Interference detection

```
Input: Threshold <$u_h$, $v_h$>;
Output: TRUE if interference detected; otherwise FALSE;
1   $X_1$ and $X_2$ are initiated once the channel has been switched;
2   On detection clock timeout
3       call Algorithm 1 to obtain the <$u$, $v$> pair;
4       $X_1 \leftarrow (1-\alpha)*X_1+\alpha*u$;
5       $X_2 \leftarrow (1-\alpha)*X_2+\alpha*v$;
6       if <$X_1$, $X_2$> is bigger than <<$u_h$, $v_h$>
7           return TRUE;
8       else return FALSE;
```

The detailed interference detecting process is presented Algorithm 2 [5]. The algorithm choose a threshold pair <u_h, v_h> as the interference reference which depends on the specific situation. As an empirical value, u_h, and v_h could be set as 20 % and −25 dBm, respectively. Then the channel will be checked periodically and <*u, v*> pair is updated at the end of every round (Line 3). After each update, the current <*u, v*> pair is compare with <u_h, v_h> and if the current <*u, v*> is higher than <u_h, v_h> (compare the first key at first and then the second key if the first key ties) (Line 6), the interference is deemed to be presented.

4.4 Static Channel Assignment in ZigBee Specification

According to the IEEE 802.15.4 specification, to form a new network, the first node —ZigBee Coordinator—scans through the list of available channels so that the network will **operate on the channel with least interference**. This means that ZigBee devices can choose a relatively clean channel by scanning all channels at the first time when the network is set up. This process is done by CCA mechanism which is introduced in Chap. 2. However, the devices cannot dynamically change their allocated channels if a predetermined channel is set to them. This is a static channel assignment scheme. Therefore, when the interference condition varies, such as node mobility and incremental WiFi deployment, they may suffer from severe interference and are exposed to performance degradation.

Figure 4.7 shows the overlapping map between ZigBee and WiFi. Depending on the WiFi system deployment, different channels could be chosen during the period of network initiation. This scheme assumes that 802.11 network occupies a fixed number of channels and the channels assigned are those left unused by WiFi. From Fig. 4.7, we can see that there exists a channel void in channel overlapping, in which there have no WiFi system deployed. Channels 25 and 26 are special cases since they do not overlap with any WiFi channel. One could argue that the interference problem would be solved simply by using channels 25 and 26.

However, this static scheme induces three aspects of problem as the following for ZigBee networks:

First, most channels are wasted if only #26 and #25 channel is used to avoid the dynamic WiFi interference.

Second, this scheme will bring out the interference between different ZigBee systems. When many ZigBee systems all use this channel to avoid WiFi interference, the interference of inter-ZigBee will become a serious problem.

Third, ZigBee networks are distributed over a larger area where interference from WiFi varies with different space.

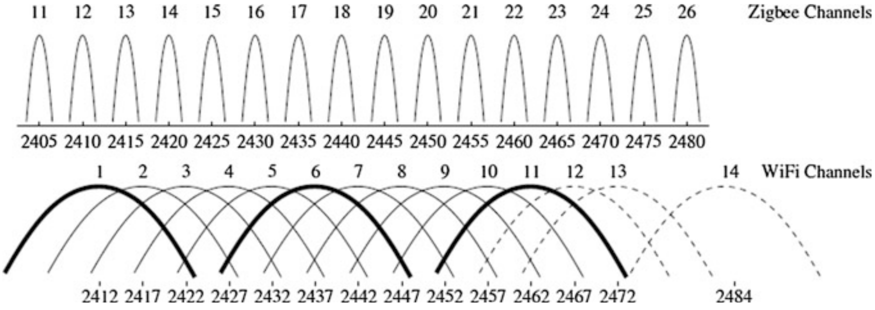


Fig. 4.7 Channel overlapping between Zigbee and WiFi

Fourth, WiFi networks in Asia and Europe occupy two more channels on the higher end of the frequency band, overlapping with Zigbee channels 25 and 26. Thus, the most dependable channel avoiding interference from WiFi is channel 26.

4.5 Dynamic Channel Assignment for Interference Avoidance

In dynamic channel assignment schemes, different nodes in a sensor network, or the same node over different points in time, will use different 15.4 channels to avoid interference from nearby WiFi sources. According to the channel numbers, the method of dynamic channel assignment can be classified into two schemes, namely single channel mode and multi-channel mode.

4.5.1 Single Channel Mode

In single channel mode, only one working channel is assigned to all ZigBee nodes to avoid the interference from WiFi. Since the interference is distributed over the networks with different intensity, the working channel need to be chosen carefully.

In order to obtain the optimal working channel, the interference information should be collected from every node so that the global interference condition could be found. To do this, every ZigBee node continuously sample the energy on each channel and send that information to a server. The server uses these monitor reports to calculate the interference contributions of each node and then runs an optimization procedure using this interference information to determine the best channel settings. The determined channel will be set on each ZigBee node.

Literature [6] is an earlier work studying interference avoidance using channel assignment. In this work, the channel assignment process has two phases: in the first phase, each of the nodes on the multihop path between the sensing node and the sink independently senses the RF spectrum to select the least noisy radio channel. In the second phase the nodes collaborate to agree on the common channel that is least congested across the whole path. Once this distributed voting phase terminates, all nodes switch to the agreed upon channel and the actual data transfer occurs. The selected channel will be used throughout the entire download operation.

Literature [7] studied the coordination on spectrum sharing between heterogeneous networks. It has the same idea but it will also set the power of the interference source, assuming all the wireless networks are controlled by a centric monitoring server.

Single channel scheme is suitable for the network with small size. With the sizes of ZigBee networks increasing, the interference condition will become more complex and the solution based on single channel will not work well any more.

4.5.2 Multi-channel Mode

Due to interference diversity in ZigBee networks, interference avoidance based on multiple channels may be more efficient than single channel mode. This subsection will discuss the solution of interference avoidance based on multiple channels.

4.5.2.1 Interference Varity and Channel Diversity

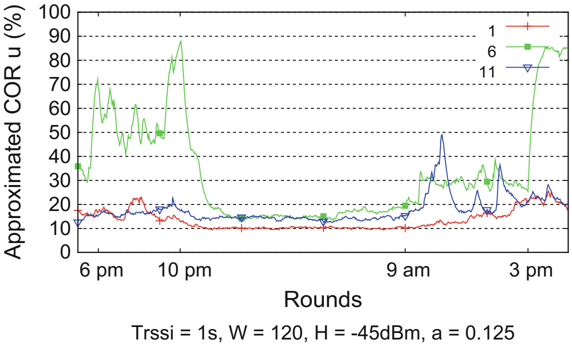
Different WiFi networks have different usage scheme and work with different channels. What is more, the working powers of different WiFi nodes are also different. In one word, there exists difference of interference from WiFi on ZigBee networks.

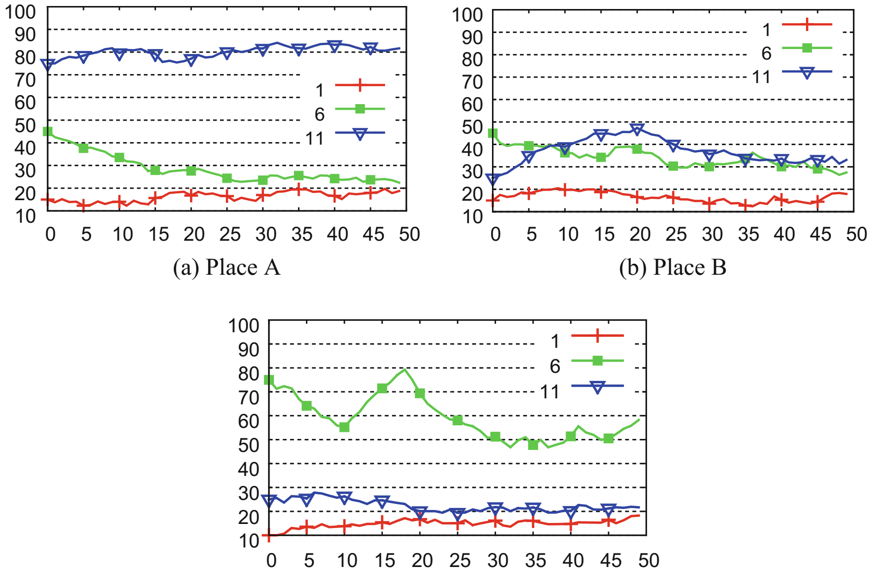
Here we present a definition which will be referred. We define the channel occupancy rate (COR) as the TIME proportion occupied by the interference signal when no ZigBee frames are transmitting. The COR is similar to the conflict time ratio, but could be easily approximated. Thus, COR can be taken as an indicator of interference density.

In order to understand the features of interference, we conducted experiments to investigate the temporal and spatial variation over different time in different space. To this end, we first used a TelosB mote randomly placed in our lab building to collect the trace of COR over time. The collected RSSIs are used to estimate the COR which shows the amount of variation. The results are plotted in Fig. 4.8 from which we can see the interference in different channels varies with different time. Specifically, the interference in channel 6 from 6 pm to 10:30 pm is stronger than that from 11 pm to 9:30 am. This is because the staffs and students were absent from the lab between 11 pm and 9:30 am, which leads to a light WiFi traffic. Thus, the predefined working channel cannot adapt with the interference if the node does not change its working channel during the whole lifetime.

Furthermore, we collect the data in three different places at the same time to investigate the interference variation with space. The results are plotted in Fig. 4.9.

Fig. 4.8 Temporal feature of interference [5]





(a) Place A

(b) Place B

(c) Place C

Fig. 4.9 Spatial variation of interference [5]

The interferences in three commonly used WiFi channel 1, 6, and 11 exhibit different features. In detail, in place A, the interference in channel 11 is the strongest among them and in place B, the interference behaviors almost in the same level. In place C, however, the strongest interference happens in channel 6. This experiment shows that the interferences are different for different places and thus the nodes in different places should work with different channels.

Literature [8] found that there is very little correlation in the quality of an IQ link across different ZigBee channels. Furthermore, when the link quality of a channel on an IQ link is bad, it is easy to find another channel where the link quality is good and it stays good typically for a few minutes. This observation is illustrated in Fig. 4.10. Besides, this figure shows that at any given point of time, it is highly likely that at least one channel is operating in the good phase. This means that when the quality of one channel is bad, the quality of the other channels can be very different and hence there is a possibility that a better channel can be found.

4.5.2.2 Multi-channel Assignment

In ZigBee specification, there are 16 channels, which provide an opportunity to exploit channel diversity to improve anti-interference ability. Each node individually scans the interference around it and set its best working channel with the least inference.

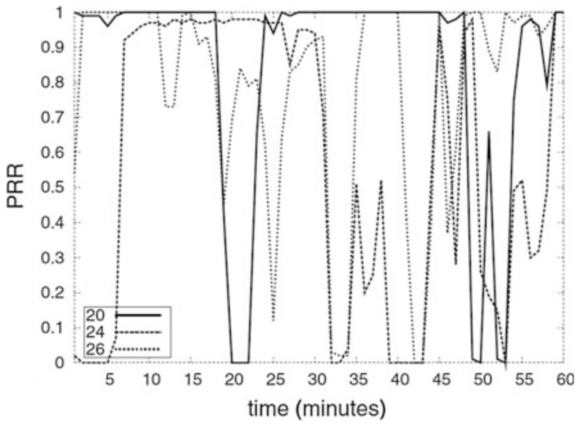


Fig. 4.10 PRR over duration of one hour for different channels [8]

Generally speaking, the interference will be minimized over the network if each node works with the cleanest channel. However, with different working channels, the nodes will not be able to communicate with each other. This will result in several "isolated island" in the networks. Therefore, there exists a trade-off between the interference avoidance and the number of working channels.

Literature [5] exploits the locality feature of interference in space to assign the working channel for the nodes located closely. These nodes that are exposed to the similar interference level will have the opportunity to own the same working channels.

To do that, a node needs to know the working channels of its neighbors. So each node maintains a ***Neighbor-Channel table*** taking <*Ngh*, *Ch*> as a table entry, where *Ngh* and *Ch* denote the node ID and its current working channel respectively. After each channel switching, the channel table will be updated responsively. With the channel table, destination channel selection algorithm works. Once interference is detected, the nodes are triggered to reassess all *M* available channels (for only one round) and obtain the newest interference indicator for each channel. The nodes will choose the quietest channel $\mathrm{CH}_{\mathrm{best}}$ as the candidate working channel.

In order to decrease the number of channels, the algorithm will search the channel to determine a channel which is used by one or several neighbors and with the closest interference level to the $\mathrm{CH}_{\mathrm{best}}$. (Note: Taking $<u_{\mathrm{delta}}, v_{\mathrm{delta}}>$ as a similar range, any channel within the scope $<u_{\mathrm{best}}, v_{\mathrm{best}}>$ to $<u_{\mathrm{best}} + u_{\mathrm{delta}}, v_{\mathrm{best}} + v_{\mathrm{delta}}>$ is viewed as the closeness) If the channel is found, it is set as the final destination channel $\mathrm{CH}_{\mathrm{dest}}$; otherwise, $\mathrm{CH}_{\mathrm{best}}$ is set as the final destination channel. Similarly, the choice of $<u_{\mathrm{delta}}, v_{\mathrm{delta}}>$ depends on the specific situation or the application demand.

4.5.2.3 Coordinating Channel Switching

After the nodes switch channel semi-dependently, some nodes working with the same channel previously could not communicate with each other anymore because the new channel may be different between these two neighbors. In the data collection service, it is a key issue to ensure the path connectivity between each node to the sink. This paper assumes that the sensor networks have formed a tree topology during the initial network deployment. Each node only needs to transmit its own data or forward its children's data to its parent. Therefore, to guarantee the connectivity in this service model, each node needs to be aware of its parent's working channel.

In order to keep the network connectivity, MuZi constructs a ***Channel-Neighbor table***, which is a reverse lookup table of ***Neighbor-Channel table***. Each table entry in ***Channel-Neighbor table*** follows the format <*Ch*, *Ngh1*, *Ngh2*,….>, where *Nghi* is denoted as a neighbor with working channel *Ch*.

For each entry, the node broadcasts to all the neighbors the information about the future working channel CH_{dest} on the neighbor's channel (Line 4) and waits for a while to collect neighbors' acks. After all neighbors' acks are received, the node switches to the new channel CH_{dest}. As such, when the node receives the broadcast message from one neighbor, it also sends back an ack and updates the ***Neighbor-Channel table***. In this way, each node is aware of its parent's working channel and just switches (if necessary) to the corresponding channel when it wants to transmit data to the sink.

4.6 Continuous Spectrum Allocation for Interference Avoidance

Traditionally, the spectrum is pre-allocated as a series of channels, each one with a fixed center frequency and frequency bandwidth. Take ZigBee channels for example, the frequency gap between two adjacent channels (with bold red segments in Fig. 4.11) cannot be used again according to ZigBee specification. These frequencies will be wasted even there exists no interference.

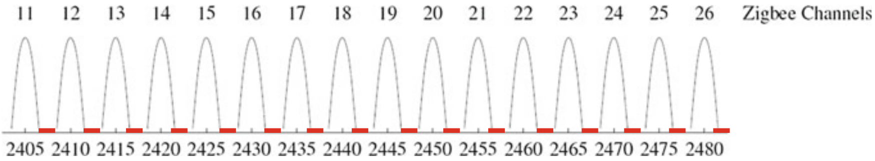


Fig. 4.11 Channel Allocation in ZigBee Protocol

Currently, these 16 orthogonal channels of the ZigBee radios are intensively explored to improve the parallelism of the transmissions. However, the interferences generated by WiFi have severely limited the usable channels for WSNs. Such a situation raises a need for a spectrum utilizing method more efficient than the conventional multi-channel access. Discrete frequency allocation artificially makes the scarce frequency unusable. If the allocation paradigm is shifted from discrete channel allocation to continuous frequency allocation, these "unusable" frequency gaps have the opportunity to be used. This strategy is helpful to improve the spectrum efficiency.

4.6.1 Feasibility of Continuous Frequency Allocation

For multi-channel based communication protocol design in WSNs, one of the most important parameters is the number of channels, which can actually be used. The commercial radio chip for ZigBee technique follows ZigBee standard and provides 16 channels. Each channel has a 5 MHz center frequency distance with neighboring channels. But the center frequency can be tuned at a granularity of 1 MHz for CC 2420, one popular ZigBee chip [9].

In literature [10], the authors study the feasibility of non-orthogonal channels by adjusting the CFD (center frequency distance) from 5 MHz to 3 MHz. Their results show that default setting of CFD = 5 MHz in ZigBee is quite conservative, smaller CFD for multi-channel design could obtain better bandwidth throughput. Assigning each channel with the channel distance that guarantees orthogonality (i.e., 9 MHz for 802.15.4) cannot fully utilize the bandwidth medium. However, if each channel is assigned with the CFD as 3 MHz, the overall throughput on a given spectrum bandwidth would improve significantly comparing to the orthogonal channel assignment scheme.

But comparing to the traditional orthogonal channel assignment, more narrow CFD introduces a critical challenge, i.e., the inter-channel interference may affect the transmission significantly. Thus, to decide a reasonable frequency distance between neighboring channels, the trade-off between larger number of channels and weaker inter-channel interference has to be carefully considered.

4.6.2 Continuous Frequency Allocation Algorithm

Although continuously allocating channel center frequencies to different links may lead to partially overlapped channels being operated simultaneously, we can combat the resulting interference by spacing them with a proper distance. In order to find the optimal frequency allocation for a set of spatially distributed nodes (or links), literature [13] proposes Frequency Allocation for Versatile Occupancy of spectrum (FAVOR) as a framework for node-level frequency allocation in for ZigBee networks.

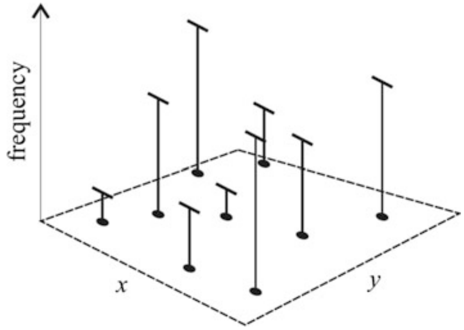
FAVOR jointly considers the frequency, distance, and continuously tuning channel center frequencies with respect to node distances. Roughly speaking, FAVOR results in a frequency allocation such that *nodes/links that are closer to each other in distance are further away from each other in frequency, while those far from each other in distance are allowed to be close in frequency*. To find an optimal frequency allocation, FAVOR creatively combines location and frequency into one space and thus transforms the frequency allocation problem into a spatial tessellation problem [11].

The problem can be modeled as follows: We assume a WSN consisting of a set of sensor nodes S = $\{n_1, n_2, \ldots, n_N\}$ with $|S| = N$, which are deployed on a 2D plane. Although FAVOR can be readily extensible to 3D volume deployments in theory, the current version is confined to 2D deployments due to the limitations imposed by the experimental conditions. Let $\{u_i\}_{i=1,\ldots,N}$ be the locations of the sensor nodes, where $u_i \in \mathbb{R}^2$. Given a frequency band $B = [f_{\min}, f_{\max}]$ and the channel width f_w, the aim of continuous frequency allocation is to assign each sensor node n_i a channel with center frequency $f_i \in B'$, where $B' = [f_{min} + f_w/2; f_{max} - f_w/2]$. Combining the node's location and frequency, a node n_i now has a new "coordinate" $(u_i; f_i) \in \mathbb{R}^2 \times B'$. We denote by $S\ (n_i)$ the one-hop neighbors of node n_i: the nodes with whom n_i can communicate directly given a common channel and a fixed transmit power.

The aim of FAVOR is to assign very different frequencies to nodes that are close to each other but arbitrary frequencies to nodes that are far from each other. The basic principle behind FAVOR can be illustrated in Fig. 4.12, where the black points indicate the sensor nodes, the "poles" on the points (with their different heights) represent different center frequencies allocated to the nodes.

From the figure, we can see that the frequency allocation is very close to the Centroidal Voronoi Tessellations (CVT) problem [11]. According to CVT model, the outcome of CVT is that the *generator*s at termination are uniformly distributed in the space, which is intuitively related to the need of frequency allocation to "spread" nodes/links over the available frequency spectrum. By define new impact metric and objective function, CVT can be applied to find a local optimal solution.

Fig. 4.12 Allocating center frequencies based on the nodes' locations



4.7 Summary

In this chapter, we discuss the methods and mechanisms in ZigBee networks for avoiding interference from WiFi. The interference detection methods are introduced from different views and some representative work on interference avoidance are discussed. Interference avoidance improves the network performance to some extent. However, interference avoidance mechanisms leave large portions of the spectrum unused even when there is little interfered traffic in them, which wastes the scarce spectrum resource. We will discuss the coexisting mechanism that enables coexistence between ZigBee and WiFi in the next chapter.

References

1. Y. Xiong, R. Zhou, M. Li, G. Xing, L. Sun, J. Ma, ZiFi: Exploiting cross-technology interference signatures for wireless LAN discovery. IEEE Trans. Mob. Comput. **13**(11), 2675–2688 (2014)
2. Y. Gao, J. Niu, R. Zhou, G. Xing, ZiFind: Exploiting cross-technology interference signatures for energy-efficient indoor localization. INFOCOM, 2940–2948 (2013)
3. F. Hermans, L. Larzon, O. Rensfelt, P. Gunningberg, A lightweight approach to online detection and classification of interference in 802.15.4-based Sensor Networks. In ACM SIGBED Review - CONET 2012, vol. 9, Issue 3, July 2012, pp. 11–20
4. F. Hermans, O. Rensfelt, T. Voigt, E.C.H. Ngai, Lars-Åke Norden, Per Gunningberg: SoNIC: classifying interference in 802.15.4 sensor networks. IPSN, 55–66 (2013)
5. G. Shi, R. Xu, Y. Shu, J. Luo, Exploiting temporal and spatial variation for WiFi interference avoidance in ZigBee Networks, special issue on: “Internet of Things”. IJSNet. **15**(4), 204–216 (2015)
6. R. Musaloiu-Elefteri, A. Terzis, Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. IJSNet. **3**(1), 43–54 (2008)
7. R. Gummadi, H. Balakrishnan, S. Seshan, Metronome: Coordinating spectrum sharing in heterogeneous wireless networks, in *First International Workshop on Communication Systems and Networks (COMSNETS)*, 2009
8. M. Doddavenkatappa, M. Choon Chan, B. Leong, Improving link quality by exploiting channel diversity in wireless sensor networks. RTSS, 159–169 (2011)
9. Chipcon’s CC2420 2.4G IEEE 802.15.4/ZigBee-ready RF Transceiver
10. X. Xu, J. Luo, Q. Zhang, Design of non-orthogonal multi-channel sensor networks, in *IEEE ICDCS*, 2010
11. Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithm. SIAM Review **41**(4), 637–676 (1999)
12. X. Zheng, Z. Cao, J. Wang, Y. He, Y. Liu, ZiSense: towards interference resilient duty cycling in wireless sensor networks. SenSys, 119–133 (2014)
13. F. Li, J. Luo, G. Shi, Y. He, FAVOR: frequency allocation for versatile occupancy of spectRum in wireless sensor networks, in *Proceedings of the 14th ACM MobiHoc* (2013), pp. 39–48

Chapter 5
Coexistence Mechanism Between WiFi and ZigBee

This chapter introduces the coexistence techniques between WiFi and ZigBee networks. Coexistence means that WiFi and ZigBee could use the same working frequency and would not jam each other. By coexistence mechanism, the low-power ZigBee networks no need to frequently adjust their working channel to avoid the interference.

5.1 Overview

Interference avoidance mechanisms try to find unused or cleaner frequency to use and thus leave large portions of the spectrum unused even when there is little IEEE 802.11 traffic. This inefficiency is especially damaging for large and dense sensor networks that cannot support the desired application throughput while using a single IEEE 802.15.4 channel.

Instead of trying to avoid interference from IEEE 802.11 traffic, the coexistence mechanism searches opportunities for ZigBee and WiFi to coexist in the same or overlapping channels. In this way, these networks can work in the same frequency and do not interrupt the transmission of other. Due to this advantage, the network resources are exploited efficiently.

Literature [1] presents a more concise definition of coexistence among low- and high-power nodes. The coexistence mechanism should meet the following goals: (a) avoiding starvation of any link, (b) avoiding significant performance deterioration of high-power links, and (c) increasing the total throughput as much as possible. In one word, the goal of coexistence is to provide some mechanism so that low-power nodes could work with high-power nodes just as they work with the same communicating technology.

However, it is not easy to pursue efficient methods to enable coexistence between WiFi and ZigBee due to the following challenges:

G. Shi and K. Li, *Signal Interference in WiFi and ZigBee Networks*,
Wireless Networks, DOI 10.1007/978-3-319-47806-7_5

(1) **Asymmetric Power levels**
ZigBee packets are transmitted with 20 dB lower power than WiFi packets, and tend to be invisible to, and often interrupted by WiFi transmitters.

(2) **Asynchronous time slots**
Even when it can be sensed by WiFi, a ZigBee transceiver has a 16 times longer response time, and is often preempted by WiFi, when it switches from sensing to transmission, or transmission to reception mode.

(3) **Heterogeneous PHY layers**
WiFi and ZigBee use incompatible PHY layers and coding/encoding techniques. Therefore, they cannot encode the packets from each other and could not coordinate the transmission via traditional MAC protocols between WiFi and ZigBee.

Currently, there are four schemes of coexistence mechanisms: *anti-Jamming redundancy coding*, *Inter-frame concurrent transmission*, *Explicit signal notification,* and *Interference Nullification and Cancellation*. We will discuss these techniques in the following.

5.2 Anti-jamming Redundancy Coding

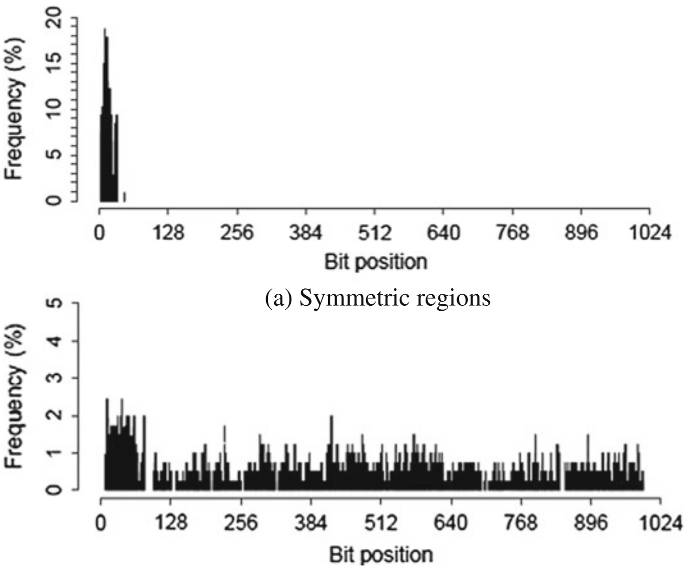
It is also well-known that there exist high packet error rate and low packet reception rate for ZigBee network when facing the transmissions from WiFi. There are many research work examined the interaction between 802.11 and 802.15.4 networks from different views. In order to recovery the error packets, *R*edundancy *Coding* is an option.

Recent work [2] examined the interference patterns between IEEE 802.11 and IEEE 802.15.4 networks at a bit-level granularity. They find that WiFi will bring out bursty bit errors in ZigBee packets in time domain due to much shorter packet length. Through extensive experimental analysis, two types of error pattern in ZigBee packets according to the interfere domain of the two radios are found:

(1) Symmetric regions: when WiFi and ZigBee transmitters are close to each other, a large percentage of dropped 15.4 packets are due to corruptions in the packet headers. This means that in this case 802.11 only corrupts the 15.4 packet header (which causes frequent packet losses), but the remainder of the 15.4 packet is left unaffected.

(2) Asymmetric regions: when WiFi and ZigBee transmitters have a long distance, interference happens in asymmetric regions. In this case, the ZigBee signal is too weak to affect WiFi frames. Compared to results in the symmetric regions, bit errors are almost uniformly located throughout the IEEE 802.15.4 packet.

These two bit error patterns under IEEE 802.11g interference source are clearly illustrated in the following pictures. The experiments are run with different distance



(a) Symmetric regions

(b) Asymmetric regions

Fig. 5.1 Bit error distribution for IEEE 802.15.4 packets when the interfering IEEE 802.11g transmitter is transmitting in symmetric region and asymmetric region [2]

between ZigBee and WiFi interference sources. The data are collected after the ZigBee sender broadcasts 2000 packets with five 802.15.4 receivers.

From Fig. 5.1, we can clearly see the difference in the bit corruption patterns. In one word, bit errors happen mainly in IEEE 802.15.4 packet headers in symmetric regions and are almost uniformly distributed in asymmetric regions. This difference implies that we need to develop different techniques, targeting individual regions, to overcome the negative impact of WiFi traffic on ZigBee transmissions.

Following this direction, Mike Liang et al. [2] provides two redundancy data transmission methods to help data recovery in different scenarios.

- **Multiple Headers**

From the experiment results we have seen that when ZigBee data transmission happens in symmetric regions, only the front section of the packet is corrupted and a large portion of packets may not be affected by interference. Therefore, if we add redundancy packet header before the original ZigBee packet, there exist uninterfered header in the received packet with high probability. In this case, multiple continuous packet headers can be used in transmitted ZigBee packets. A two-headers packet structure is illustrated in Fig. 5.2 where the detailed meanings can be found in Chap. 2.

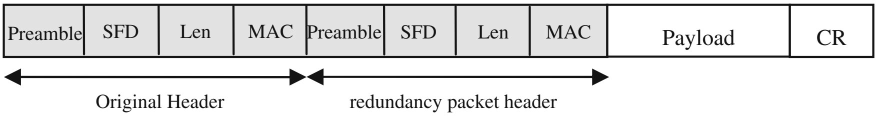


Fig. 5.2 Frame structure with two headers

When using multiple headers, if there is no interference from WiFi, the ZigBee radio treats the second packet header as a part of the payload. As processed in ZigBee specification, the receiver will detect the first SFD after the first preamble sequence and will treat this SFD as the start of the packet. On the other hand, if the first preamble bytes are corrupted due to WiFi transmission, the ZigBee receiver may not detect the first SFD correctly. However, in symmetric regions, the second packet header has a high uninterrupted probability and the receiver will continue to detect the second SFD preceded by four preamble bytes and assume a new packet starts. Therefore, the header-corrupted packet can also be encoded in the receiver.

However, this methods need to modify the ZigBee protocol stack. For example, in order to pass CRC check, the standard CRC filtering on the radio needs to be disabled. At the same time, the radio software stack is also modified to include a 2-byte CRC within the packet's payload. This CRC covers the innermost header (excluding the length field) and the user payload. And also the receiver's software stack needs to remove any remaining headers from the payload before delivering the packet to the user application.

- **Forward Error Correction**

When bit errors are uniformly located throughout the received packet, multiple headers is not useful again because each header will have the opportunity to be corrupted.

FEC augments each packet with extra information that enables the receiver to correct some of the bit errors. Although FEC algorithms are widely used, selecting the right algorithm and implementing it on resource-constrained sensor nodes are nontrivial. Literature [2] implemented a full-featured Reed Solomon library for resource-constrained devices. The FEC coding/encoding is run in MAC layer of ZigBee nodes to increase the likelihood of successfully decoding the packet at the destination for multi-hop transmission.

The RS code divides a message into x blocks of user-defined size and computes a parity of y blocks. An RS-encoded message then consists of the original message and the computed parity. The length of the parity determines the maximum number of corrupted and erasure blocks in the encoded message that the receiver can successfully recover from.

5.3 Inter-frame Concurrent Transmission

Intuitively, WiFi will not always occupy the channel which is affected by many factors such as backoff due to signal collision and user traffic characteristic. Many experiment results reveal that the network traffic is highly bursty and the channel is free in most of time. Figure 5.3 shows a typical trace of channel usage of the same WiFi network. We can see that WiFi traffic leaving significant amount of white spaces between 802.11 frames. The frames intervals are caused by the backoff durations and some special intervals (DIFS and SIFS) built-in WiFi transmissions. Due to the bursty and clustered WiFi traffics, many small idle leaks within the frame clusters and large ***white space*** (defined as >1 ms idle duration) between the frame clusters are left.

Ideally, according to CSMA/CA mechanism, ZigBee nodes could have the chance to transmit their data by sensing the channel status during these white spaces. However, Literature [3] found that standard CSMA in ZigBee is ineffective in utilizing the white space left by WiFi due to the *heterogeneous PHY layer* and *power asymmetry*. With this condition, even ZigBee transmitters can sense WiFi transmitters, but not vice versa. *As a result, WiFi transmitters cannot sense ZigBee signals and hence do not defer their transmissions even when there exist ongoing ZigBee packet transmissions. Therefore, WiFi signals can easily corrupt the ongoing reception of ZigBee packets. Hence, it is necessary to make improvement on ZigBee transmission.*

For transmitting packets in frame intervals, transmitting time taken by the packets from ZigBee nodes cannot exceed the length of frame intervals. However, WiFi frames are clustered together with short intervals typically less than 1 *ms* while the minimum packet transmission time of ZigBee, after accounting for the software overhead, approaches the maximum backoff window size of 802.11. Therefore, it is very difficult for ZigBee senders to utilize the short WiFi frame inter-arrival times for packet transmission. Thus, how to efficiently utilize the frame interval is a key problem.

5.3.1 *Frame Length Adaptation in White Space*

WISE proposed in [3] *predicts the length of white space in WiFi traffic and intelligently adapts frame size to maximize the throughput efficiency.* Based on a Pareto

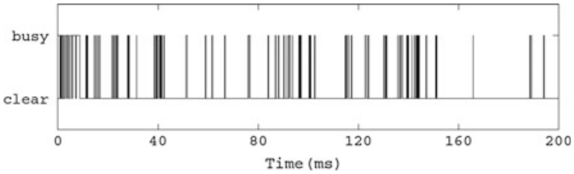


Fig. 5.3 WiFi channel state trace

model for the size of idle duration in WiFi channel, ZigBee devices can transmit packets with appropriate size to ensure a low collision probability. *The basic process is explained in the following*:

1. ***Traffic Features***: Huang et al. [3] *found by analyzing the collected WiFi trace that WiFi traffic has a time-scale invariant feature, which suggests that the arrival process of WiFi frame clusters is self-similar. The detailed self-similar features can be found in subsection IV.B in* [3].
2. ***Traffic Modeling***: *Based on this founding, some random process models are used to model the* distributed inter-arrival time *of WiFi frame clusters. Pareto process is one of the most widely used model to fit the power law distributions which is a* feature of self-similarity.
3. ***Frame Size Adaptation***: Based on the traffic model, the length of white space in WiFi traffic could be predicted. To limit the collision probability with WiFi traffic, MAC frame sizes are split adaptively to match the remaining lifetime of the white space using the Pareto model.

5.3.2 Coding Adaptation in White Space

Frame size adaption in white space is a collision-avoidance scheme trying to access WiFi channel with size-reduced ZigBee packets to reduce the collision probability. However, as most idle duration sizes are too small, there are not abundant opportunities for successful ZigBee transmissions especially for those when WiFi channel *is busy.*

The efficiency of forward error correction (FEC) coding has been verified in [2] *and results show that FEC technique is effective in recovering corrupted ZigBee packets in WiFi interfering environment. However, Literature* [2] *uses a fixed coding technique for ZigBee device. This means the redundancy ratio remains constant despite of the bit error rate. However, the channel conditions are affected by highly dynamic WiFi interference. Therefore, it is inefficient for ZigBee to utilize the channel with dynamic WiFi traffic using fixed coding rate.*

As we all know, different FEC coding techniques exhibit significant performance differences which has a high influence on the link throughput of ZigBee. Considering limited storage resource and low computation ability of ZigBee devices, dynamically adjusting coding schemes and selecting appropriate FEC coding techniques based on channel condition could improve link throughput.

Literature [4] *conducted performance test of 3 Hamming and 3 BCH coding techniques and proposed a coding adaptive coexistence methods*:

(1) *Based on the WiFi traffic model, the distribution of WiFi interfering time in a ZigBee packet is estimated. Then, through measuring the Signal to Noise plus Interference power Ratio (SNIR), the distribution of the number of error bits in a ZigBee packet can be calculated.*

(2) *Thus, given a FEC coding technique, the probability that the ZigBee packet fails be recovered can be estimated. In this way, the transmission efficiencies of the FEC coding techniques in different WiFi traffic cases are derived.*
(3) *Taking transmission efficiency into account, ZigBee select the optimal coding strategy based on their measurement on the current channel.*

For real-time adaptation in data transmission, adaptive encoding rules table referred by a sender are first built based on the average inter-arrival time of WiFi frame clusters, the average size of WiFi frame clusters and RSSI. The table can be precalculated offline. By real-time collecting, the current information of these three parameters, the sender derives the models and decides the encoding technique for the next packet according to the encoding rules, thus achieving real-time adaptive transmissions.

However, these methods cannot guarantee ZigBee's performance during busy WiFi traffic, since WiFi is still the "first-class citizen." Second, it introduces extra overhead to existing ZigBee protocols and needs reprogramming existing nodes, which is difficult to be applied in existing deployed architecture [5].

5.4 Explicit Signal Notification for Coexistence

Basically, MAC protocol in network stack bears the responsibility for coordinating channel access. However, the function cannot work between *heterogeneous* networks because they could not encode the packets from each other. More importantly, it is difficulty for WiFi to sense the existence of ZigBee due to the low transmit power. But if we can bring some signals which could be "heard" by the high-power network, WiFi will sense the presence of ZigBee *and thus some cooperation opportunities helpful to solve the blind competition over channel could be created.* The mutual visibility solutions can enhance a fair coexistence. *Currently, there are three ways to emitting this explicit notification signals to enhance the visibility of ZigBee to WiFi.*

5.4.1 Explicit Notification Using Hybrid Device

Explicit notification signal is needed to be understood by the receiver. However, one standard cannot communicate with the other without modification to the underlying hardware. Literature [6] proposes a solution which utilizes a dedicated communicating node whose setup includes both ZigBee and 802.11 transmitters. This node has the ability to transmit both 802.11 and ZigBee messages and thus has the gateway function. The goal of augmenting the device is to temporarily block out 802.11 messages for a window of time large enough that ZigBee devices can successfully transmit their messages, thereby resolving the interference issue.

With the help of this hybrid device, when a ZigBee nodes want to transmit, this hybrid device could transmit an 802.11 packet indicating that this packet would have an unusually long duration (perhaps 64 ms or so). Other WiFi nodes hearing this message will be blocked during this period so that the low-power ZigBee has a clear channel to transmit its data leisurely. This method is very easy to be implemented by modifying the duration filed in 802.11 frame body, but has a danger of abusing 802.11 protocols. Literature [6] found that the 802.11 devices completely ignored the rogue packets.

There is another way for this hybrid device to intervene between 802.11 and ZigBee. That is to use RTS/CTS messages to clear 802.11 traffic. The hybrid device sends out a CTS before ZigBee nodes want to transmit data. CTS message will block all 802.11 devices from transmitting for a specified period of time.

If 802.11 traffic can be blocked for short periods of time which are long enough for ZigBee devices to transmit, then it will be possible to develop a market solution that will alleviate the contention issues between ZigBee and 802.11. With the ability of communication with both ZigBee and WiFi devices, the arbitrator can schedule ZigBee and WiFi's activities without collision. However, the coordination between the signaler device and other ZigBee devices is not considered.

5.4.2 Explicit Notification Using ZigBee Device

Under the consideration of cost, augmenting dedicated devices to emit WiFi signal is not efficient. A better solution is to directly utilize ZigBee node to transmit WiFi-aware signal.

In [7], authors proposed to use a special ZigBee device (called signaler) to send busy tone concurrently with the desired data transmission in the area where ZigBee and WiFi coexist. *ZigBee cannot issue a packet to the 802.11 devices indicating that it wishes to transmit data based on current* modulation and packets types and so forth. Therefore, the signaler is designed with high transmission power so that the signal from the device could be detected by WiFi easily. In this way, WiFi is forced to backoff its transmission during the period of ZigBee transmission. The architecture is illustrated in Fig. 5.4.

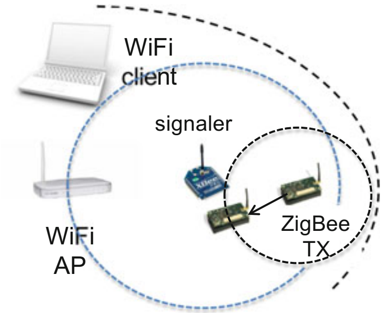


Fig. 5.4 Using signaler to send busy tone

For efficiently using this method, there are four key points:

(1) **Avoiding WiFi preemption**: In order to preserve the channel and prevent WiFi preemption, the busy tone must be started before the actual data transmission and carrier sensing and keep speaking during the whole process of data transmission.

(2) **Busy tone failure**: Due to their disparate power levels, ZigBee signals may not be effectively sensed by WiFi, where ZigBee can hear WiFi, but WiFi is oblivious of ZigBee. To handle this issue, some ZigBee nodes close to WiFi interferers should work as the signaler, by transmitting a busy tone synchronously, thus notifying WiFi to suspend its transmission.

(3) **Avoiding interfering transmitter**: In order to prevent the signaler from interfering with the transmitter, transmitter sends data packet on some channel while the signaler hops to an adjacent channel to send the busy tone. In this way, the busy tone can be heard by WiFi, but is orthogonal to the data packet.

(4) **Synchronization with Data**: In order to synchronize the busy tone and data transmission, the signaler initiate each transmission. It performs CCA and backoff just as a normal ZigBee node and broadcasts a notification message (referred to as CTS) to other nodes if the channel is clean and then switches to the adjacent channel to start emitting the busy tone. At this time, the clients will finally contend for the channel access similarly to legacy ZigBee after they receive the CTS message.

5.4.3 *Explicit Notification Using Customized Preambles*

As introduced in Chap. 2, we have seen that a WiFi packet transmission begins with a PHY preamble, followed by a PHY header, and then the DATA. There exists a LENGTH filed in the PHY header which specifying the number of microseconds that WiFi packet lasts. According to 802.11 protocols, the WiFi radio will refrain from transmitting for 2^LENGTH microseconds if the radio detects a PHY preamble. Based on this observation, Wang et al. [8] proposed to send a fake WiFi PHY header by setting LENGTH as the temporal length of ZigBee active interval to refrain WiFi from transmitting. After sending the fake WiFi-compliant signal, a temporal white space will be created for ZigBee to communicate.

Different with fake WiFi PHY, Weeble proposed in [9] exploits the phenomenon of signal detection that longer preamble sequence can be detected easier [10]. Based on a key observation, the preamble of ZigBee wireless technology is redesigned. The length of the new preamble is adaptively tuned based on additive-increase–multiplicative-decrease (AIMD) principles using packet losses as a feedback. The preamble has the role of signaling a start of a low-power reservation to a high-power node. Any ZigBee node can signal a start of a transmission reservation period. Once a period has started all WiFi nodes that heard the preamble will refrain from transmitting for the period of time. After that, all ZigBee nodes nearby will find the medium idle from high-power transmission and may contend for the access using CSMA for the duration of the period.

The above solution make low-power ZigBee nodes send WiFi-ware single from different view and provide an efficient way to handle the interference. However, it is not a perfect solution because long-term running of mutual visibility solutions will cause WiFi performance degradation, and WiFi can also have anti-jamming capability to make such signals infeasible. Moreover, the signaler solution requires strict timing control of ZigBee's transmission, leading to severe protocol overhead in large-scale ZigBee networks.

5.5 OFDM Subcarrier Nulling

In recent years, orthogonal frequency division multiplexing (OFDM) modulation has moved out of textbooks and research laboratories and into practice in modern communications systems. OFDM divides a given channel into many narrower subcarriers or bins. The spacing is such that the subcarriers are orthogonal, so they will not interfere with one another despite the lack of guard bands between them. IEEE 802.11a/g uses a form of OFDM which creates 64 subcarriers.

Compared with ZigBee, 802.11a/g is wideband networks. They have a wider working spectrum. If the strong WiFi devices first find the existence of weak ZigBee devices and decide which spectrum ZigBee networks use, we can utilize OFDM subcarrier suppressing or nulling technique to vacate spectrum that ZigBee networks are using. By subcarrier suppressing we mean that those subcarriers overlapping with ZigBee working spectrum will not be used for data transmission any more.

Subcarrier suppressing means ensuring that no power is used in bins marked as narrowband-occupied. OFDM naturally allows different power assignments for each frequency bin. A straightforward realization way for subcarrier nulling in IEEE 802.11 OFDM PHY is presented in [11]. For nulling one subcarrier, the transmitter can simply feed 0's to the subcarriers instead of sending information bits (1 or −1) as normal transmission in OFDM. This operation will result in zero power on the corresponding spectrum. Thus, little interference or no interference is generated to ensure proper operation of low-power devices, which enables the coexistence of different technologies.

Literature [11] studies partial spectrum sharing in wireless LANs and proposes adaptive subcarrier nulling. Adaptive subcarrier nulling technique groups the subcarriers into several subbands, and allows neighboring WLANs to share and contend for access to each subband. When a shared subband is occupied by one WLAN, another WLAN can opportunistically null the corresponding subcarriers in that subband, and use those nonoverlapping subbands to send packets. This idea is can be illustrated in Fig. 5.5. Figure 5.5a illustrates that a WLAN often needs to share part of its spectrum with others where the channel widths are heterogeneous. Figure 5.5b shows that subcarrier nulling can opportunistically null the

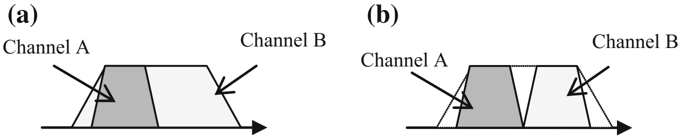


Fig. 5.5 **a** Normal spectrum-overlapping transmission. **b** Adaptive subcarrier nulling nulls the shared busy subband (containing a number of subcarriers) and leverages the nonoverlapping subbands to send data

corresponding subcarriers of Channel B in that subband overlapping with Channel A. In this case, the band width of original Channel B will be decreased correspondingly.

Although Literature [11] shows it applicability only for multiple WLANs, this scheme can be applied to handle the coexistence between WiFi and ZigBee if the strong WiFi devices could find the existence of weak ZigBee devices either by sensing or learning. In this way, some continuous subcarriers of WiFi are nulling to allocate these frequency bands to the low-power ZigBee nodes.

Literature [12] is the first system enables high-throughput wideband nodes to coexist with unknown narrowband devices. When sharing its spectrum with the narrowband nodes, the frequency will be fragmented into many noncontiguous bands. This phenomenon is very common since a wide contiguous unused band typically does not exist especially for ISM frequency bands. Therefore, a PHY layer that can operate over noncontiguous spectrum bands must be provided so that the PHY on the receiver side can receive in noncontiguous bins even when narrowband devices are using the other bins

From above introduction we can see that though subcarrier nulling or suppressing solution is a good idea to provide general coexistence, it requires special hardware redesign on high-power nodes, e.g., preamble design and packet detection algorithm, which limits the application if it is not compatible with existing devices [5]. What's more, the high-power nodes must have the ability to detect the existence of low-power nodes and their working frequency bands. It is not very easy for WiFi to detect the information when ZigBee nodes locate far from WiFi nodes.

5.6 Interference Nullification and Cancellation

Above coexistence methods need either to modify protocol stack or to augment dedicated devices, which are involved tremendous work. For coexistence, all nodes in either ZigBee or WiFi networks should update or modify their protocol stack or running program. Furthermore, the above coexistence mechanism will case more or less performance degradation. A more attractive coexistence solution should do less modification while make two systems share the conflicted network resource, without harming each other.

5.6.1 Basic Idea of Interference Cancellation

Successive Interference Cancellation (SIC) [13] is a well-known physical layer technique. Traditionally, only the strongest signal can be decoded when two or more packet transmissions arrive at a receiver simultaneously and the other signal is treated as interference. However, SIC facilitates recovery of even the weaker signal. For this, the bits of the stronger signal are decoded as before. The original (stronger) signal is then reconstructed from these bits, and subtracted (i.e., cancelled) from the mixed signal. The bits of the weaker packet are then decoded from this residue. This can be an iterative process to recover multiple packets and hence it is termed successive interference cancellation. The flow diagram is depicted in Fig. 5.6. Briefly, SIC enables a receiver to receive two or more signals concurrently (that otherwise cause a collision in today's systems). Thus, SIC is a useful tool for dealing with diverse power transmissions.

5.6.2 ZigBee Signal Recovery Using IC

Literature [14] proposed a sink-only-based solution named ZIMO combining with the architecture of wireless sensor networks. The overall processing in ZIMO is based on SIC. Obviously, the signal strength of WiFi is always 5–20 dB stronger than that of ZigBee due to high transmit power. For recovering the low-power ZigBee signal from WiFi and ZigBee mixed signal, SIC is a proper choice.

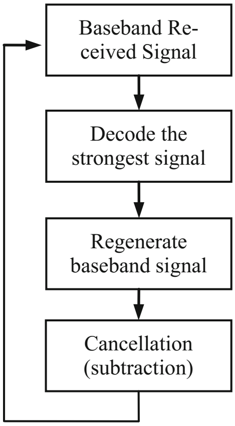

Baseband Received Signal
Decode the strongest signal
Regenerate baseband signal
Cancellation (subtraction)

Fig. 5.6 Flow diagram of interference cancellation

According to the SIC technique, ZigBee signal can be first regarded as background noise, and it is possible to first decode WiFi packet by standard decoder given enough SNR of WiFi signal. Subtracting the strong decoded WiFi signal by interference cancellation (IC) technique, WiFi signal can be mitigated from the mixed signal. Then the standard ZigBee decoder can extract ZigBee packets from the left signal.

In order to demultiplexing the interested signals from the mixed signals, ZIMO designed a MIMO-based sink with two antennas. When WiFi preamble is clear, WiFi signal is directly nullified and the residual signal is left for ZigBee decoding. However, if WiFi preamble is not clear, ZIMO uses the clear ZigBee preamble to nullify ZigBee signal first to decode WiFi signal. After that using the interference cancellation technique, the WiFi interference is mitigated, and the residual signal can be used for ZigBee decoding. In this way, WiFi data and ZigBee data can be recovered simultaneously.

In order to implement this method, a wide-band RF frond-end is needed to perform wideband sampling so that both WiFi and ZigBee information is not lost. The sampled signal is converted to suitable signal for ZigBee decoding. This method only modifies the sink and thus avoiding reprogramming all the deployed sensor nodes. However, this method requires total control of wireless physical layer, which cannot be accomplished using commercial network interface cards and sensor nodes [14]. Thus, it is not easy to implement all features in ZIMO.

Literature [5] further proposes a method WizBee to decode the collided WiFi or ZigBee signal with only one antenna in the sink. WizBee has the same motivation and metrology with ZIMO and is compatible with current ZigBee and WiFi system. But WizBee further design downlink data transmission from sink to sensor end nodes. Downlink design is also accomplished by only modifying WizBee sink node. In WizBee, there is not any protocol modification. The only requirement is to replace conventional ZigBee sink with WizBee sink.

Above solutions leverage the crosstechnology signal relationship between WiFi and ZigBee in time, spectral, and power domain and recovers the ZigBee and WiFi signal simultaneously in sink node from received interfered signals. Since it only modifies the sink node, they provide for coexistence a relatively easy and more affordable solution.

5.7 Summary

In this chapter, we discuss the coexistence solutions for WiFi and ZigBee. These solutions are motivated by different aspects and exploit different mechanisms in ZigBee and WiFi protocols. These methods shield a light on performance improvement of ZigBee. However, there are still a long way for efficient coexistence between ZigBee and WiFi. We believe that with more new techniques developing, more solutions will emerge in the future.

References

1. B. Radunovic, R. Chandra, D. Gunawardena, Weeble: enabling low-power nodes to coexist with high-power nodes in white space networks, in *CoNEXT* (2012), pp. 205–216
2. C.-J. Mike Liang, B. Priyantha, J. Liu, A. Terzis, Surviving wi-fi interference in low power ZigBee networks, in *SenSys* (2010), pp. 309–322
3. J. Huang, G. Xing, G. Zhou, R. Zhou, Beyond co-existence: exploiting WiFi white space for Zigbee performance assurance, in *ICNP* (2010), pp. 305–314
4. P. Guo, J. Cao, K. Zhang, X. Liu, Enhancing ZigBee throughput under WiFi interference using real-time adaptive coding, in *INFOCOM* (2014), pp. 2858–2866
5. Y. Yan, P. Yang, X.-Y. Li, Y. Zhang, J. Lu, L. You, J. Wang, J. Han, Y. Xiong, WizBee: wise ZigBee coexistence via interference cancellation with single antenna. IEEE Trans. Mob. Comput. **14**(12), 2590–2603 (2015)
6. J. Hou, B. Chang, D.-K. Cho, M. Gerla, Minimizing 802.11 interference on ZigBee medical sensors, in *BodyNets* (2009)
7. X. Zhang, K.G. Shin, Enabling coexistence of heterogeneous wireless systems: case for ZigBee and WiFi, in *MOBIHOC* (2011)
8. Y. Wang, Q. Wang, Z. Zeng, G. Zheng, R. Zheng, WiCop: engineering WiFi temporal white-spaces for safe operations of wireless body area networks in medical applications, in *RTSS* (2011), pp. 170–179
9. B. Radunovic, R. Chandra, D. Gunawardena, Weeble: enabling low-power nodes to coexist with high-power nodes in white space networks, in *ACM CoNEXT* (2012)
10. S.M. Kay, in *Fundamentals of Statistical Signal Processing*, Detection Theory, vol 2 (Prentice Hall, Upper Saddle River, 1998)
11. X. Zhang, K.G. Shin, Adaptive subcarrier nulling: enabling partial spectrum sharing in wireless LANs, in *ICNP* (2011), pp. 311–320
12. H. Rahul, N. Kushman, D. Katabi, C. Sodini, F. Edalat, Learning to share: narrowband-friendly wideband networks, in *Proceedings of ACM SIGCOMM* (2008)
13. D. Halperin, T. Anderson, D. Wetherall, Taking the sting out of carrier sense: interference cancelation for wireless lans, in *Proceedings of ACM MOBICOM* (2008), pp. 339–350
14. Y. Yan, P. Yang, X.-Y. Li, Y. Tao, L. Zhang, L. You, ZIMO: building cross-technology MIMO to harmonize zigbee smog with WiFi flash without intervention, in *MOBICOM* (2013), pp. 465–476

Chapter 6
Cooperation and Communication Between WiFi and ZigBee

The preceding chapters mainly focus on the mechanism and solution of communication competition between WiFi and ZigBee. But on the other hand, the same frequency band occupied by both ZigBee and WiFi enables their radios to sample background energy so that they can sense the transmission activities from another. This overlapping frequency across WiFi and ZigBee provides collaboration possibilities in some specific environments. Currently, many solutions are proposed to coordinate heterogeneous devices without modifying their PHY layer modulation schemes. We will discuss in this chapter the mechanism and system for cooperation and communication between WiFi and ZigBee.

6.1 Motivation

This section presents some motivations or benefits for cooperation between WiFi and ZigBee.

6.1.1 Energy Conservation

WiFi radio interface in mobile devices is attracting an increasing amount of applications ranging from mobile social networking to mobile localization. However, WiFi interface consumes a considerable amount of power when active compared with ZigBee node due to its higher transmission power. There exist many cases that a WiFi radio has to stay active without performing any real communications such as waiting for incoming packets, scanning new AP, etc. If the operations of a WiFi radio can be delegated to a ZigBee radio with lower power so that the WiFi radio could be turned off when there is no packet to transmit and receive, the significant energy consumptions on WiFi radio are reduced to the reasonably

G. Shi and K. Li, *Signal Interference in WiFi and ZigBee Networks*,
Wireless Networks, DOI 10.1007/978-3-319-47806-7_6

low-power consumptions on ZigBee radio. Motivated by this, there are some attempts using a ZigBee radio to do WiFi operations. It is attractive for mobile devices powered by battery.

6.1.2 Capability Supplement

WiFi and ZigBee were designed for different environments and have different features in terms of power and energy. The protocols from PHY to application layer have different working mechanism and procedures. Due to cost and complexity, one communicating technology does not have the necessarity to implement the mechanism that required of another.The two technologies possess strengths in different areas that are often the weaknesses of the others. Therefore, both networks can be enhanced via mutual supplementation, demonstrating the positive side of coexistence. Motivated by the collaborating potential, there are some representative works which emerge recently and achieve good effect.

6.1.3 Cross-Technology Communication

Currently, there are a variety of standards in the same spectrum, especially for ISM frequency band. Enable communication between devices with fundamentally different physical layers is important. Many application scenarios need cross-technology communication ability, such as coordinated coexistence, network management, and coordination in Internet of things (IoT) [1, 2]. Cross-technology communication is used for communication between devices belonging to the same standard by sensing the signal energy from others.

6.2 Cooperation Between WiFi and ZigBee

This section discusses the how to cooperate between WiFi and ZigBee so that the network performance is improved.

6.2.1 Obtaining Cooperation from WiFi

6.2.1.1 Using WiFi for ZigBee Clock Synchronization

There are many fundamental services for wireless sensor network required for many applications. Time synchronization, making the node to maintain a common notion

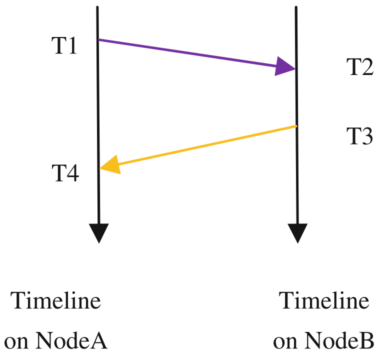


Fig. 6.1 Basic idea of time synchronization based on message passing

of time, is one of the examples. Many applications of WSNs require the nodes to maintain a common notion of time such as sleep schedule and data aggregation. A number of protocols are proposed based on message passing. The basic idea is illustrated in Fig. 6.1. When clock synchronization begins, node A send a message with time stamp of $T1$ and node $T2$ receives the message at time $T2$. Then node A will receive a message from node B with a time stamp $T3$ at time $T4$. Time stamp $T1$, $T2$, $T3$, $T4$ are all referred to timeline on the local node and are known by node 1 when it receives the second message. Thus we have the following two equations: $T2 = T1 + \text{delay} + \text{offset}$ and $T4 = T3 + \text{delay-offset}$. Therefore we can calculate the time offset between node A and node B as $(T2 - T1) - (T4 - T3)/2$. As result, the clock will be synchronized and the accuracy depends on how to attach the time stamp on the message.

However, message passing for clock synchronization will incur high massaging overhead for large-scale WSNs. What's more, the synchronization error or clock skew will grow with network size. If the nodes can be synchronized via an external global timebase, such as global positioning system (GPS), the message overhead and clock skew will be reduced considerably especially for large-scale wireless sensor networks. However, using GPS will require hardware update for receiver to decode the out of band clock signal, introducing extra cost and design complexity.

WiFi networks have been widely deployed in the past years and will continue to enjoy a phenomenal penetration rate in our life. For coexisting ZigBee, WiFi could be considered as another "GPS" if it can emit time signal so that the to-be-synchronized ZigBee nodes can hear it. Fortunately, the IEEE 802.11 standards require all WiFi access points (APs) to broadcast periodic beacon frames for the purpose of network management. Working on the same radio frequencies, ZigBee sensors can detect the transmissions of such beacons and use them as a clock signal to synchronize their clocks [3]. Using the same clock synchronization source, the clock skew can be corrected timely.

However it is not easy to obtain the time information directly from WiFi periodical beacon because ZigBee cannot decode the packets from WiFi. What's more, the transmission of beacon frame (default beacon period in 802.11 is 102.4 ms) may be delayed due to channel contention caused by pending or ongoing data transmissions. Hao et al. [3] shows by experiments that the beacon period jitters does not keep constant and will vary with time and traffic overload. Thus, the beacons with high jitter must be removed before used to calibrate the clock.

Literature [3] use the method proposed in [4] to extract the beacons after it discards the outliers in the detected beacons. The remaining beacons yield high periodicity and are used as a reference clock as input for time synchronization. The method of beacon extraction has been introduced in Chap. 4. Since we are not focusing on time synchronization of wireless sensor networks, the detailed algorithm will not be discussed in this book.

6.2.1.2 Using WiFi for ZigBee Location

The location problem in sensor networks has equally important place with time synchronization and m is one of the reasons that prevent sensor networks from becoming more prevalent. Currently, there have been many literatures reporting indoor localization systems to handle with sensor localization problem. The basic method is to infer the sensor location via pre-deployed anchor nodes whose location is manually measured. The premise is that the sensors are able to receive and understand the message from anchors.

WiFi AP, largely installed in indoor environments, has the same role as anchor node if we can get the signal such as RSSI from AP. Their locations are managed by a network system manager. Using WiFi APs as anchors, we can localize sensor nodes without newly deployed anchor nodes. The only problem is to obtain the RSS of each AP and then calculate the sensor's location using the relationship between RSS and distance.

To measure RSS of WiFi signals on sensor nodes, literature [5] developed a cross-technology signal extraction scheme. This scheme employs the signal folding technique discussed in Sect. 4.2.2 of Chap. 5. It detects the AP signal by periodic beacon signal and then retrieves AP-RSS with a simple filtering method is obtained. With the RSSI information of each AP, the sensor locations can be calculated based on the signal attenuation model.

6.2.2 Obtaining Cooperation from ZigBee

6.2.2.1 Using ZigBee for Beacon Detection

WiFi discovery is an important mechanism for WLANs. IEEE 802.11 provides two scanning methods: active and passive. During an active scan, the client radio transmits a probe request and listens for a probe response from an AP. With a passive scan, the client radio listens on each channel for beacons sent periodically by an AP. Active discovering new WiFi APs wastes the precious energy of mobile devices due to excessive scanning operations of WiFi network interface cards.

Motivated by that ZigBee radio can detect WiFi signal by energy sensing, ZiFi [4] uses low-power ZigBee radios to identify periodical WiFi beacons, thus discovering WiFi networks. ZiFi enables ZigBee radios to collect and detect the signatures with unique interference patterns created by WiFi beacons. When the client losses the connection with WiFi AP, the WiFi radio will be turned off and discovering operations on ZigBee nodes are triggered. Only when detecting the existence of WiFi APs in a purely passive manner, the WiFi NIC will be woken up.

In order to find WiFi existence on ZigBee nodes, ZiFi searches in RSS samples the beacon signal periodically broadcasted from AP. ZiFi uses a novel digital signal processing algorithm to identify periodic signals from the sampled RSS series. The algorithm has been extensively exploited by consequent works. We have introduced the basic idea in Chap. 4 and do not discuss it again here.

6.2.2.2 Using ZigBee for Energy-Efficient Internet Access

Widely deployed WiFi networks in urban areas have become an important way of Internet access for mobile devices such as mobile phone. It is well known that WiFi is still a big energy consumer compared to other components; even new phones have shown great improvements. Turning WiFi radio off when there is no traffic is the most efficient solution to reduce the energy consumption. However, continuous reachability is required for some applications on the devices. Therefore, the WiFi must be always kept on for management message exchange such as registration and paging [6]. Generally speaking, these messages are of shorter packets length and hence an amount of energy has to be wasted for mobile phones during the period of waiting the management messages. If these management messages can be offloaded from WiFi to other low-power interfaces, the WiFi interface can be turned off for energy saving.

Based on the motivation, [6] proposes to utilize a dual WiFi-ZigBee radio on mobile phones and APs as an auxiliary link to bear the low rate data transmission. They design and implemented the system WiZi-Cloud by augmenting a ZigBee interface on the mobile devices and AP nodes. With this dual-radio communication device, the traffic can be demultiplexed to different interfaces so that the WiFi have the chances to be turned off. Therefore, the network stack of the AP is extended to maintain connectivity with the mobile devices through the ZigBee interface (e.g., beaconing and paging for ZigBee), as well as to coordinate with peer APs to locate mobile devices.

Its basic mechanism can be illustrated in Fig. 6.2. In one word, WiZi-Cloud use additional ZigBee radios to help WiFi clients achieve high energy efficiency while keeping ubiquitous connectivity and real-time inter-AP handover. This is an example of collaboration between WiFi and ZigBee, benefit to WiFi.

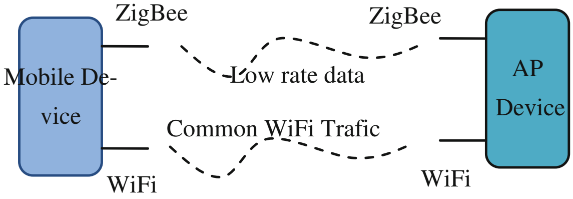


Fig. 6.2 Dual-radio coordination for internet access using ZigBee

6.2.2.3 Using ZigBee for WiFi Monitoring

Wireless networks monitoring is to detect the network status to help users to find various performance issues. WizNet [7] further proposes to use ZigBee-based Sensor System for monitoring WLAN. It adopts the digital signal processing proposed in ZiFi to identify and associates 802.11 signals with the corresponsive AP based on RSS measurements. For performance monitoring, WizNet uses the measurement data to estimate the channel utilization rate and throughput.

In performance monitoring, signal strength is an important indicator. However, the RSS sampled in ZigBee nodes only measures a fraction of power due to its narrower bandwidth compared with WiFi. Energy out of the current channel cannot be detected. To accurately reflect the signal quality of 802.11 transmissions, WiZNet periodically changes the center frequency of sampling nodes, which enables ZigBee to sample the signal from a much wider bandwidth.

In detail, WiZNet have implemented the following function [7]: First, the manager running on a PC collects a small amount of information about beacon frames logged by WLAN APs, and then jointly processes them with the beacon RSS measured by sensors through a cross-correlation algorithm. Second, after associating RSS samples with APs, WizNet manager estimates the SNR of each AP and the channel utilization rates at monitoring spots. Finally, the manager estimates the throughput between local WLAN clients and the monitored APs, and detects rogue APs. Therefore, WiZNet have the following key designs:

AP identity Association: A cross-correlation algorithm is used to associate each AP with the RSS measurements of its beacons by computing the dot product between the two RSS traces obtained by sensors and APs, respectively.

SNR and channel utilization: WizNet manager first calculates the sensor SNR by subtracting the base noise of the ZigBee sensors from the RSS samples. The base noise is computed by applying exponential moving average over the minimum values in the RSS series. Then the manager infers the SNR of APs that a WLAN client would receive at every monitoring spot.

Rogue AP Detection: Since the RSS measured by sensors are associated with the APs using cross-correlation, WizNet manager labels each identified RSS in the series obtained by sensors with BSSIDs of the APs. As a result, any AP that cannot be identified is potentially a rogue AP.

6.2.2.4 Using ZigBee for Radio Map Building

A WiFi radio map [8] is useful to many applications such as AP selection and localization. It shows the signal strength distribution of WiFi APs over different locations in a given environment. Radio map just likes a database recording the signal distribution information. With a radio map, the AP that transmits stronger RSS can be selected to provide better quality.

Traditionally, to build a WiFi radio map, a mobile device with a WiFi network interface card (NIC) are controlled to scan all channels for RSS at each location to collect the signal information. However, this approach is limited due to its time-consuming way of sweeping the entire space.

With the increasing deployment of sensor networks in various sectors, WiFi networks coexisting with ZigBee networks are common. ZigBee networks are distributed over the spaces with a higher density and ZigBee radio can also sense WiFi signals although it cannot decode WiFi frames. Therefore, using ZigBee to build the radio map will reduce the time consumption.

WiBee [8] builds real-time WiFi radio maps using a ZigBee networks. The ZigBee sensors collect the RSS information and send back to a gateway node which creates the map. The created map can be accessed by all WiFi clients. The gateway node is a dual radio device combined WiFi interface and ZigBee interface. With the help of the gateway, WiBee works as follows [8].

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

Using the same methodology, ZIL [9] proposes to use ZigBee interface to collect the RSS for fingerprint-based location. In the fingerprint-based localization process, the signal fingerprint or the radio map have to be pre-collected in training process and used for matching with real-time fingerprint in testing phases. To achieve real-time localization, WiFi-enabled devices have to constantly scan the WiFi channels, resulting in high power consumption and reduced battery runtime. Contrary to this, ZIL used ZigBee interface to take the place of WiFi interface for mobile client to detect the WiFi fingerprint for matching the location of the client using fingerprint. All frequently scanning operations are offloaded from WiFi interface to ZigBee interface. Basically, this solution is based on ZiFi [4] which is proposed to find whether there exists WiFi AP by detecting the beacon message using ZigBee.

6.3 Communication Between WiFi and ZigBee

Traditionally, two nodes can communicate only if they can decode the packets from each other. Thus the traditional communication happens on the packets level and only works between the same physical layers. However, in a broader sense, when the signal is sensed by energy detection, communication happens! Even if devices have fundamentally distinct physical layers, they can still communicate with each other through energy packet sensing. Furthermore, if the sensed energy could be transformed into a set of code regularly, cross-technology message transmission between heterogeneous networks by sensing the energy becomes possible. By sensing the energy patterns on the channel, simple message can be transmitted. But obviously, it cannot provide high-rate communication ability. The following context will present the representative works.

6.3.1 Esense

Cross-technology communication through energy sensing is first studied in [10] which proposed an energy communication framework Esense. Esense establishes communication channels from WiFi to ZigBee by modulating packet lengths to those are unlikely to be used in normal WiFi traffic.

For transmission message via energy, a base alphabet is needed and a set size of two (corresponding to bits 1 and 0) is a minimum requirement for building a vocabulary (i.e., sequences of alphabets). Here alphabet is much like a symbol used at the PHY layer. Suppose the alphabet set that could be built based on practical considerations is of size M, each alphabet basically conveys $\log_2 M$ information bits. However, the achievable transmission rate R of Esense is $R = \log_2 M/(A+B)$ where A and B are the channel access delay and average packet length, respectively [10]. This means that larger the alphabet set, more the information that can be conveyed (numerator), but the average transmission time can also be high.

For constructing the alphabet, a ZigBee radio is used to continuously measure the background energy. In this way, a certain number (denoted as #+consec) of consecutive positive energy samples (i.e., samples with energy readings greater than a certain threshold) will be generated. Using data analysis of several public WiFi traces, Esense studies the distributions of #+consec and proposes that those #+consec that rarely appear can form an alphabet and each character in the alphabet can represent a different piece of information conveyed from WiFi radios to ZigBee radios.

In Esense system, a message (sequence of bits) is mapped to an alphabet and a WiFi packet corresponding to that size is transmitted (at a predetermined rate). This transmission will result in an appropriate energy burst duration, which can be sensed by the receiver and mapped to the appropriate alphabet and, hence, corresponding bits. Through measurement of WiFi traffic traces from real-world

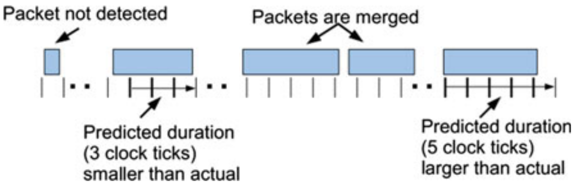


Fig. 6.3 Possible problem due to the hardware limitations of ZigBee [10]

deployments, the result of bimodal distribution of packet length is found. Then for determining the alphabet set, a solution is to exclude all packet sizes whose frequency of occurrence in the WiFi traces is greater than a threshold percentage and the rest is allocated as alphabets for Esense.

The above is the basic idea for energy-based communication. However, in reality, it is very difficult to accurately measure the length of energy burst due to the limitation of ZigBee hardware and multiple 802.11 data rates. Some possible problems for detection of packet length are illustrated in Fig. 6.3 where some packets are not detected due to having enough time and some packets are merged due to the very small gap. For addressing the above issues, [10] present the corresponsive solutions, such as restricting that the Esense packets are always sent at the lowest possible data rate at the sender, sending the Esense alphabet (packet) multiple times in a small time-window and choosing a margin to ensure that a gap between any adjacent alphabet exists. Meanwhile, the difference between two adjacent predefined message packet sizes should be set appropriately to ensure ZigBee will not generate the same number of energy samples for message packets with different sizes. But anyway, the accuracy depends on multiple factors and its performance has to be proved in real system.

6.3.2 HoWiES

For energy saving, HoWiES [11] proposes to use the side channel to transmit information from wireless AP to ZigBee radios in mobile devices whose WiFi can be turned off. When ZigBee decodes the information transmitted through the side channel, it can either ignore the message if it is not for the device, or wake up the WiFi radio for communication.

In order to make ZigBee understand the messages from WiFi, HoWiES extends the Esense mechanism to convey data with combinations of WiFi packets. Its main improvement on the communicating technique is described in the following.

Similar to ESense, the alphabet is built on the predefined packets sizes. But HoWiES use the combinations of the characters to form different messages, instead of letting each character in the alphabet correspond to a piece of information. Assume the messages that WiFi radios can deliver to ZigBee radios correspond to different numbers. That means that different number represents different message that a WiFi radio wants to convey to a ZigBee radio. Then when WiFi radio wants to send message to ZigBee radio, it encodes the number by sending a sequence of WiFi packets (called WiFi-ZigBee message packets), whose sizes are chosen from a group of predefined values, using a fixed transmission rate. These predefined packets sizes form the alphabet of our message delivery scheme. The ZigBee radio determines the size of each packet by sampling background energy, and obtains the number that the WiFi radio wants to convey by interpreting the combination of packet sizes.

Denote the alphabet A as a set of b packet sizes: $A = \{S_1, \ldots, S_\mathrm{b}\}$, where $S_1 < \cdots < S_b$ and have a corresponsive WiFi to ZigBee packet with a predefined length. In order to ensure that the receiver ZigBee radios can distinguish out a WiFi-ZigBee message, the message packets must be different with normal WiFi packets. When beginning a message transmission, the sender's WiFi Radio encodes a message M by sending a sequence of l message packets, whose size are chosen from the alphabet A.

For example, if the alphabet A is $\{100, 200\}$, the size of the alphabet b is 2. When WiFi radios encode each message by transmitting three WiFi packets, these packet lengths are chosen from 100 and 200 bytes. In this case, the message length l is 3. The total number of messages that a WiFi radio can convey to a ZigBee radio is $2^3 = 8$. If a WiFi radio encodes a message by sending a sequence of three packets with 200B, 100B, and 200B, respectively, essentially it sends out three digits with values of 1, 0, and 1 in that order, and the message is interpreted as number 5 (i.e., $1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 = 5$). This method extends the capacity of ESense.

6.3.3 GSense

Like other communication based on energy sensing, GSense [12] creates a side channel also between heterogeneous wireless devices. GSense also wants to leverage the capability of energy sensing to coordinate heterogeneous devices without modifying their PHY layer modulation schemes. To achieve this, GSense prepends legacy packets with a customized preamble that contains multiple energy pulses. The preamble leverages the quiet period between signal pulses to convey coordinate information if the preamble can be detected by neighboring nodes even when they have incompatible PHY layers.

In order to be detected by a heterogeneous receiver, the gap between signal pulses must be sufficiently long. What's more, the energy pulse should not be too long; otherwise, the receiver may confuse it with legacy packets. Suppose the transmitter's sampling clock-rate is D times that of the receiver's (WiFi always has a higher sampling rate or spectrum width than ZigBee), then each energy pulse must contain at least D samples, such that at least one of them can be received. The minimum gap length must be larger than D, the ratio of spectrum width between the transmitter and the receiver, and shorter than the carrier sensing duration of neighboring CSMA devices so that the gap may not be taken as a hint for an idle channel to avoid the colliding with subsequent energy pulses.

From the above introduction, we can see that there are too many circumscribes in designing an efficient impulse sequence. It is obvious that GSense is only suitable for delivering low rate control/coordination information [12]. Suppose the maximum gap length in GSense is G_m samples, then up to $\log_2 G_m$ bits of information can be carried in each gap. If the sampling rate is F Hz, the resulting information capacity is $\log_2 G_m/(G_m/F) = (F \log_2 G_m)/G_m$ bits/s. The capacity increases linearly with sampling rate F, but decreases with gap length. Hence, a wider gap may waste more channel resources, albeit more information bits can be delivered.

6.3.4 FreeBee

FreeBee [1] is based on 802.11 and 802.15.4 standards, and compatible with all 802.11 variants (i.e., a/b/g/n) and 802.15.4-compliant nodes. Most important, it requires no hardware modification and does not introduce dedicated traffic. Above methods have no this feature.

FreeBee cleverly uses the periodic beacon from 802.11 AP and shifts the beacon time from its original position along the timeline to indicate the symbol to be delivered. The basic idea is illustrated in Fig. 6.4 where the original position is at t, beacon interval is T. FreeBee assumes that the original position of the beacon is found during the initial network setup, which we refer to as the reference position. FreeBee represents the information by shifting the beacon time in the range of $(-T/2, T/2]$. The information is embedded in each time shift and the amount of information that can be embedded is determined by T and the granularity of shift, denoted by Δ.

We have known that a typical T is 102.4 ms for 802.11 protocols. If Δ is set as 1.024 ms (which is compliant to the beacon scheduling granularity in the 802.11 standard), there exist 100 Δs, indicating that the beacon can be positioned at 100 different time instances. Thus, there are $\lfloor \log_2 100 \rfloor = 6$ bits that can express by beacon time shift.

Like other energy-based communication, FreeBee demodulation process is also based on energy sampling in the channel by consecutively recording the RSSI. Further apply the *folding* technique proposed in [4] (we present its idea in Chap. 4), the reference position of the beacon and the information beacon can be find. Then the difference of the two positions indicates the symbol. If the difference has been encoded beforehand, the communication between two the sender and the receiver success.

However, in order to demodulate the information carried by the shifted beacon, many times transmissions are needed, which will have a negative effect on the

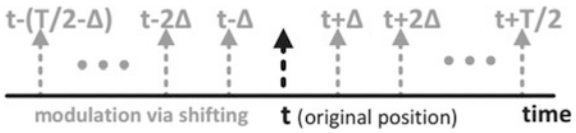


Fig. 6.4 Encoding time shift of beacon message for communication

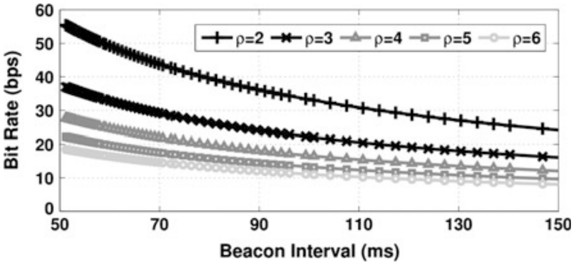


Fig. 6.5 The theoretical bit rate for FreeBee

performance. Denote as ρ as the number of beacon repetitions required for obtaining the difference, bit rate R of FreeBee can be computed as below. Figure 6.5 shows the impact of beacon interval T on R in different scenarios for the range of practical intervals where Δ is 1.024 ms

$$R = \frac{\log_2 T/\Delta}{T \times \rho} \text{bps}.$$

References

1. S.M. Kim, T. He, FreeBee: cross-technology communication via free side-channel, in *MobiCom* (2015), 317–330
2. K. Chebrolu, A. Dhekne, Esense: communication through energy sensing, in *MOBICOM* (2009), 85–96
3. T. Hao, R. Zhou, G. Xing, M. Mutka, Wizsync: exploiting Wi-Fi infrastructure for clock synchronization in wireless sensor networks, in *Proceedings of IEEE RTSS* (2011), 149–158
4. Y. Xiong, R. Zhou, M. Li, G. Xing, L. Sun, J. Ma, ZiFi: exploiting cross-technology interference signatures for wireless LAN discovery. IEEE Trans. Mob. Comput. **13**(11), 2675–2688 (2014)
5. S. Ishida, K. Izumi, S. Tagashira, A. Fukuda, WiFi AP-RSS monitoring using sensor nodes toward anchor-free sensor localization, in *IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, (Boston, MA, Sep 2015), pp. 1–5
6. T. Jin, G. Noubir, B. Sheng, WiZi-cloud: application-transparent dual ZigBee-WiFi radios for low power internet access, in *Proceedings of IEEE INFOCOM* (2011)
7. R Zhou, G Xing, X Xu, J Wang, L Gu, WizNet: a ZigBee-based sensor system for distributed wireless LAN performance monitoring, in *PerCom* (2013), pp. 123–131
8. W. Li, Y. Zhu, T. He, WiBee: building WiFi radio map with ZigBee sensor networks, in *Proceedings of IEEE INFOCOM* (2012)
9. J. Niu, B. Wang, L. Shu, T.Q. Duong, Y. Chen, ZIL: an energy-efficient indoor localization system using zigbee radio to detect wifi fingerprints. IEEE J. Sel. Areas Commun. **33**(7), 1431–1442 (2015)
10. K. Chebrolu, A. Dhekne, Esense: energy sensing-based cross-technology communication. IEEE Trans. Mob. Comput. **12**(11), 2303–2316 (2013)
11. Y. Zhang, Q. Li. Howies: a holistic approach to zigbee assisted wifi energy savings in mobile devices, in *INFOCOM* (2013)
12. X. Zhang, K.G. Shin, Gap sense: lightweight coordination of heterogeneous wireless devices, in *INFOCOM* (2013)