Enhancing Zigbee throughput under WiFi interference using real-time adaptive coding

Peng Guo and Jiannong Cao
Hong Kong Polytechnic University
Hong Kong, China
Email: {cspguo, csjcao}@comp.polyu.edu.hk

Kui Zhang
University of Twente
Netherlands
Email: K.Zhang@utwente.nl

Xuefeng Liu
Hong Kong Polytechnic University
Hong Kong, China
Email: csxfliu@comp.polyu.edu.hk

***Abstract*—Co-existing in the unlicensed ISM band, ZigBee transmissions can be significantly interfered by WiFi. Although several approaches recently are proposed to enable ZigBee transmission under WiFi interference, the ZigBee throughput still decreases to zero when WiFi throughput (generated by D-ITG) is over $8Mbps$. In this paper, we propose a real-time ($< 5ms$) adaptive transmission (RAT) scheme to efficiently adapt forward error-correction coding (FEC) on ZigBee devices in dynamic WiFi environment. We find that sizes of WiFi frames well follow the power law distribution model. With the model, corruption in ZigBee packets can be estimated to some extent, thus facilitating ZigBee device to choose a suitable FEC coding to maximize the throughput. Extensive experimental results show that, compared with existing works, RAT achieves significant performance improvement of ZigBee transmissions in WiFi environment with different traffic load. Particularly, the ZigBee throughput of RAT can be about $10kbps$ when the WiFi throughput is $8Mbps$.**

I. Introduction

In the last few years, there has been an increasing adoption of ZigBee technology for low-power, cost-effective, flexible, reliable, and scalable wireless products. These products are driven by the large number of emerging applications including game remote controllers, health care monitoring, industrial automation, and so on. Many of these applications are performance-sensitive, where high link throughput or low delivery delay is required. For instance, in health care monitoring, wireless tags attached on patients' body must reliably report cardiac rhythm data at desired rates. In game remote controls or industrial automation, command delivering delay should satisfy user experience or typical demands.

However, the emerging ZigBee applications today are facing severe challenges with the proliferation of WiFi hotspots. Most of existing ZigBee applications work in indoor environment, e.g., home, hospital or coffee shop, which is usually covered by WiFi APs. As these two wireless communication technologies operate in almost the same unlicensed ISM band, there is serious interference between each other [1]. Typically, due to the low transmission power level (e.g., $0dBm$), ZigBee transmissions usually suffer from the interference of WiFi (over $20dBm$) while WiFi cannot even sense the existence of ZigBee's signal. As a result, ZigBee device always has to suspend its transmissions when WiFi is active, which severely limits the performance-sensitive ZigBee applications.

To improve the ZigBee transmission performance in WiFi environment, a straightforward approach is to assign orthogonal frequency channels to ZigBee and WiFi devices. However, as the ISM band can be partitioned with only 3 orthogonal WiFi channels, multiple WiFi APs working in different channels occupy almost the entire ISM band in WiFi hotspots, remaining very limited collision-free spectrum for ZigBee.

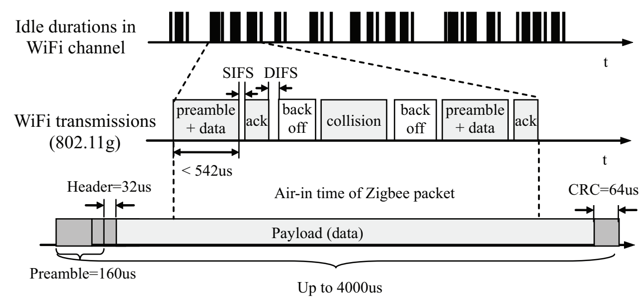


Fig. 1: Idle durations in WiFi channel and air-in time of ZigBee packet.

People find that there are abundant idle duration in WiFi channel in time domain [2]. However, efficiently utilizing the idle duration for Zigbee transmission is not a trivia. The key reason is that the size of idle durations in WiFi channel is usually much smaller than that of the air-in time of ZigBee packet. Fig. 1 gives an illustration of idle durations of WiFi channel (802.11g) and the air-in time of ZigBee packet. It can be seen, when ZigBee device detects the idle channel, its packet may still suffer from WiFi interference.

To facilitate ZigBee packets to be successfully transmitted in WiFi channel, there are two typical approaches: **collision-avoidance** based approach [3] and **collision-recovery** based approach [4]. The collision-avoidance approach tries to access WiFi channel with size-reduced ZigBee packets. Based on a Pareto model for the size of idle duration in WiFi channel, ZigBee devices can transmit packets with appropriate size to ensure a low collision probability. However, as most idle durations sizes are too small, there are not abundant opportunities for successful ZigBee transmissions, although the ratio of idle duration in WiFi channel can be still high. Also, the reduction of packet size decreases the transmission efficiency (valid dada within the transmission time), with requiring more packet headers.

For the collision-recovery based approach, some forward-error-correction (FEC) coding techniques are applied on ZigBee devices. When a small part of an encoded packet is collided, the packet can still be recovered. However, selecting a right coding technique for resource-constrained ZigBee devices is non-trivial. Many existing coding techniques either have high storage requirement or have a much high time complexity when running on ZigBee devices. For instance, when decoding a RS-encoded packet with TelosB devices, the time is over $50\times$ time for transmitting the packet [4].

In this paper, we exploit the collision-recovery idea for ZigBee transmission. Particularly, we propose a *real-time adaptive encoding* idea, where ZigBee devices can quickly (¡5ms) and appropriately adapt the coding according to the highly dynamic WiFi traffic. The proposed *real-time adaptive encoding* has following difference from traditional adaptive encoding solutions:

1) First, traditional adaptive encoding (or rate adaptation) usually aims at handling the slow variety of link quality (e.g., due to the weather or movement) [17], [18]. Characteristics of the variety due to external interference are seldom addressed. Hence, under the highly dynamic WiFi interference, traditional adaptation cannot timely follow the quick variety of the link quality.
2) Second, most traditional adaptive encoding schemes do not consider the encoding/decoding time which actually affects the link throughput. It may not be a problem for communication terminals with high computation ability, such as WiFi terminals. However, it becomes significant for ZigBee devices which have very limited computation ability. Few existing works study the performance of coding on ZigBee devices except for [4].

To implement the *real-time adaptive encoding* idea, obtaining the real-time WiFi interfering traffic model with ZigBee device is a key issue. The work in [3] makes one step on it, which accurately models the *white space* in WiFi channel. However, to achieve the goal, some problems still need to be addressed: 1) can the duration when WiFi occupies the channel also be modeled? 2) can ZigBee update the model while transmitting packets? 3) how to choose appropriate coding techniques for the resource-limited ZigBee devices? This paper gives the positive answers to these questions, as well as an explicit solution which can be easily followed in practice.

Our motivation is based on two observations from existing extensive WiFi data traced in different scenario and our own testing data: 1) both the size of WiFi frame clusters and the size of *white space* between the clusters in practice well satisfy the power law distribution; 2) there is a distinct duration gap between ZigBee's encoding time and decoding time, which can be sufficient for channel sampling and model updating. Leveraging these two observations, it is possible for ZigBee to update the model while transmitting packets without extra sampling time. Based on this motivation, in this paper, we first give the theoretically analysis of the distribution of errors in ZigBee packet with our model. Then, a series of light-weight FEC coding techniques with different error-correction abilities are tested on ZigBee-based TelosB motes. With the testing result of the encoding/decoding time and the errors distribution, we derive the optimal encoding strategy for ZigBee devices in different WiFi traffic scenario. Meanwhile, we design an optimal *segment-based CRC* mechanism to improve the packet delivery probability. After that, we present the real-time adaptive transmission (RAT) scheme, with introducing the details on how ZigBee obtains the real-time channel parameters in practice. Numerous experiments are conducted, and the results show the remarkable performance of our proposed scheme compared to related works mentioned above.

Contributions of the paper are summarized as follows.

1) We conduct extensive statistical analysis of data traces captured in real-life WiFi networks. We show that, in a channel shared by a group of WiFi devices, the size of WiFi frame clusters well satisfy the power law distribution.
2) We test the encoding/decoding time of a series of light-weight coding techniques on ZigBee devices. Based on the results and the WiFi traffic model, we derive the transmission efficiency for each coding technique, and present the optimal encoding strategy for different WiFi traffic load.
3) We observe that there is distinct duration gap between encoding time and decoding time on ZigBee devices, which can be effectively exploited for channel sampling. Hence, ZigBee devices can keep updating the WiFi traffic model while delivering packets, without frequently suspending their transmissions for channel sampling.

The remainder of the paper is organized as follows. Section II reviews related works. Section III introduce the motivations of the proposed scheme. In Section IV and Section V, the WiFi traffic model and tests of FEC coding techniques on ZigBee devices are described, respectively. In Section VI, we present the real-time adaptive transmission scheme based on the WiFi traffic model and testing results. Numerical experimental results are presented in Section VII, followed by the conclusions made in Section VIII.

II. Related Work

Coexistence has long been a problem for protocols operating on the ISM band. Several studies have been conducted using analysis and simulation to evaluate the effect of WiFi on ZigBee [5], [6]. Many experiments under WiFi environment also have been made on ZigBee devices [7], [8]. Both simulation and experiments results show the significant degradation of ZigBee performance in the presence of WiFi interference.

To avoid the interference, The IEEE 802.15.2 [9] proposed an adaptive frequency hopping (AFH) mechanism for ZigBee devices so that they can dynamically find a temporal collision-free channel in ISM band. However, AFH incurs substantial overhead to ZigBee devices, as they need to scan the entire 16 channels and re-establish connections with neighbors. This problem becomes more pronounced in a dynamic network with mobile WiFi devices [10].

Another idea for ZigBee to avoid the WiFi interference is opportunistic channel access in time domain. A ZigBee channel randomly occupied by WiFi leaves much separated white space that can be exploited by ZigBee devices. This idea has been extensively discussed in the field of cognitive radio (CR) [11]–[13]. However, the secondary user in CR, which opportunistically uses the channel, can access the channel only when primary users (i.e., licensed users) do not use it, and must release the channel after a dedicated duration in order to not interfere with the possible primary users. While, in the scenario of the coexistence of ZigBee and WiFi, ZigBee devices do not need to consider the affection on WiFi, as they share the unlicensed ISM band. Therefore, methods in CR are not suitable to the problem discussed in this paper.

To efficiently exploit the white space of the channel, [3] proposed a transmission mechanism for ZigBee devices based on a statistical model for the *white space*. In the mechanism, the size of ZigBee packets is reduced to be an appropriate value, so to minimize the corruption due to WiFi interference. However, the reduced size of packets decreases the transmission efficiency, leading to increment of the communication overhead. In addition, when the channel is busy with WiFi's activities, ZigBee devices have to keep suspending their transmissions, resulting in poor performance for the delay-sensitive applications.

Some other people consider placing a special hybrid device (called arbitrator or jammer) to help the low-power ZigBee fairly share the channel with WiFi [14], [15]. With the ability of communication with both ZigBee and WiFi devices, the arbitrator can schedule ZigBee and WiFi's activities without collision. However, in dynamic networks, the scheduling becomes much complex because any node's movement would require the coordinator to re-initiate a spectrum survey and re-allocate the parameters. In [16], authors placed a special ZigBee device (called signaler) in the area where ZigBee and WiFi co-exist. Due to the high transmission power, the signaler device can be detected by WiFi with sending busy tone, thus forcing WiFi to back off its transmission. In this way, the signaler device can help other ZigBee devices to access the channel. However, the coordination between the signaler device and other ZigBee devices is also complex and unreliable under the WiFi environment.

Recently, [4] proposed $BuzzBuzz$ protocol to improve the coexistence of ZigBee and WiFi devices operating in the overlapping frequency channels. Two techniques are employed in the $BuzzBuzz$. One is to set multiple packet headers for ZigBee, which can make some types of WiFi devices have to backoff their transmissions thus saving the data part of the ZigBee packet from the interference. However, this technique will not work for the WiFi devices with packet detection based CCA mode. Another technique employed in $BuzzBuzz$ is the forward-error-correction (FEC) coding. Experimental results show that FEC coding technique exhibits remarkable effectiveness in recovering corrupted ZigBee packets in WiFi interfering environment. However, as $BuzzBuzz$ protocol employs a fixed coding technique for ZigBee device, the protocol is inefficient for ZigBee to utilize the channel with dynamic WiFi traffic.

In order to efficiently apply FEC coding technique in dynamic channel, adaptive encoding approach is usually employed in digital communication [17], [18]. The approach adapts the coding rate according to the channel conditions that is generally provided in terms of signal-to-noise ratio (SNR). It mainly helps to mitigate for slow channel fading mostly caused by rain attenuation. However, in ZigBee channel interfered by WiFi traffic, the channel condition is mostly affected by the highly dynamic WiFi interference, which is a very different case.

III. Preliminary

We assume that a ZigBee network exists in the communication range of multiple WiFi ($802.11b/g$) APs and terminals. The number of APS and terminals is unknown. We consider two kinds of traffics on ZigBee nodes: data flow (e.g. the traffic in health monitoring) and command delivery (e.g. the traffic in remote control). We expect to design a media access control (MAC) scheme for the ZigBee devices, which implements real-time adaptive transmissions with the WiFi interference, so as to achieve high link throughput on the ZigBee links for data flow traffic and low average deliver delay on ZigBee links for command delivery traffic in the WiFi collision environment.

The motivation of the real-time adaptive transmission with WiFi interference is based on following three issues.

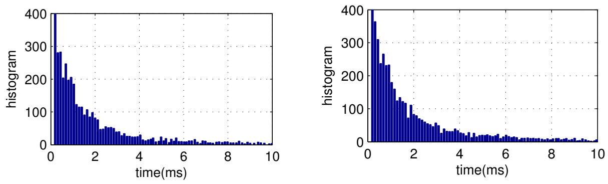
1) Although sizes of most idle intervals in WiFi channel are far smaller than the air-in time of ZigBee packets, in view of the low ratio of busy duration to idle duration in WiFi channel, data in ZigBee packets can still successfully delivered when a light-weight FEC coding technique is employed.
2) Besides existing observation that the arrival time of real-life WiFi frame clusters can be modeled by Pareto distribution, we further observe that the size of WiFi frame clusters can also be accurately modeled by a power law distribution.
3) Moreover, we find that there is a significant difference between the encoding time and decoding time of FEC coding technique when applied on the resource-limited ZigBee device, and the difference of time can be sufficiently utilized for real-time WiFi traffic modelling.

Based on the motivation, in the following Sections, we first introduce the method of modeling real-life WiFi traffic and performance tests of a series of light-weight FEC coding techniques on ZigBee devices. Then, we deduce the optimal encoding strategy and describe the implementation details of the real-time adaptive transmission scheme on ZigBee devices in practice.

IV. Model the Real-life WiFi Traffic

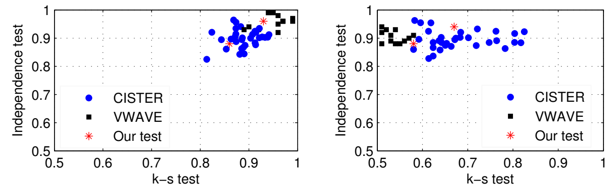
In order to estimate errors in ZigBee packets corrupted by WiFi interference, we model WiFi activities in the shared channel. Existing works have shown that, WiFi traffics are usually bursty and clustered [3], leaving many small idle leaks within the frame clusters and large *white space* (defined as $> 1ms$ idle duration) between the frame clusters. The leaks are associated with the back-off durations and some special intervals (DIFS and SIFS) built-in WiFi transmissions, while the *white space* is the long interval between WiFi traffics ordered by the upper layer.

In this paper, we further conduct the size of WiFi frames with extensive statistical analysis on data traces captured in real-life WiFi networks. We show that, in a channel shared by a group of 802.11 devices, the size of aggregate WiFi frame clusters has the feature of heavy tail distribution. We then study



(a) CISTER trace (b) VWAVE trace

Fig. 2: Histogram for WiFi frame cluster.



(a) Window size = $100ms$ (b) Window size = $500ms$

Fig. 3: Goodness-of-fit tests of power law model for real-life WiFi traces.

in what time scale the WiFi frame clusters are modelable and present a power law model that accurately characterizes the size.

First, we give an analysis on the size of WiFi frame clusters in practice. With the same definition in [3], a WiFi frame cluster is an independent set of sequential WiFi frames that have less than $1ms$ intervals between each other, and the interval between any two frame clusters is larger than $1ms$. We analyze the open data traces in CISTER 2012 [19] and VWAVE 2009 [20]. The former trace is captured in library building with ZigBee nodes detecting WiFi's RSSI, while the latter trace is captured in six typical environments (e.g., office building, library, cafeteria, coffee shop) with a commercial sniffer appliance VeriWave monitoring the air-in WiFi packets. Both of the two data traces contain the size information of WiFi frames.

Fig. 2 gives the statistical results of WiFi frame cluster's size with the data files from CISTER 2012 and VWAVE 2009. The Y-axis shows the number of clusters with the corresponding size in a data file. From Fig. 2, the distribution of the clusters' size has the typical heavy-tail feature. To characterize it, we establish a following power law distribution model.

$$Pr\{x \leq t\} = \begin{cases} 1 - (\frac{1}{1-\alpha+t})^{\beta} & \text{if } t > \alpha, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

where α and β are the scale and shape of the power law model. According to our observation, we set α to $0.2ms$. In power law model, β is given by $\lambda_c/(\lambda_c - \alpha)$, where λ_c is the average size of WiFi frame clusters.

To evaluate the goodness-of-fit of the power law model, Kolmogorov-Smirnov Test (K-S test) is used, which is a widely adopted tool to test the goodness-of-fit. We divide the time into equal sized windows. For the frame clusters in each window, a power law distribution is fitted by maximum likelihood estimation. K-S test is then applied for each window to test the goodness-of-fit for the estimated power law distribution. We set the significance level as 0.9. For each window, K-S test outputs a decision whether the window passes the test. In the power law model, we also assume that the sizes of WiFi frame clusters are independent of each other. To test this assumption, we compute the one lag autocorrelation for each window [21].

Fig. 3 gives the results of goodness-of-fit test on the power law model. We conduct K-S test on two data traces which are captured in CISTER 2012 and VWAVE 2009. Each data trace includes a group of trace files, which are different in the captured time, the monitoring channel, and the position of the traffic sniffer. In addition, we also collected data of WiFi activities with TelosB motes in Starbucks coffee shop for 8 hours, and get two data files for the detection in the morning and afternoon, respectively. For each file in the two data traces and our collections, we check the goodness-of-fit with different window sizes. Due to space limitation, we only show the results of $100ms$ and $500ms$. Each trace file corresponds to two points in the figure, while each of our data file corresponds to one point. The x-value of the point is the percentage of windows in the corresponding data trace file that pass the K-S test, when the window size is set to a specific value. The y-value is the percentage of windows that pass the independence test. We expect that, if points are clustered at the top right corner, then the fitness of the power law model is good. We observe that the modelability of the frame cluster size varies with time scale. At a small time scale of $100ms$, the cluster sizes can be well characterized by the power law model.

TABLE I: Performance of light-weight coding methods

Coding method	Tolerant errors $\frac{t}{n}$	Valid data	Encoding time	Decoding time
BCH(15, 1)	$7/15$	$7B$	$706\mu s$	$83.9ms$
BCH(15, 5)	$3/15$	$30B$	$249\mu s$	$74.8ms$
BCH(15, 7)	$2/15$	$53B$	$247\mu s$	$54.7ms$
Hamming(7, 4)	$1/7$	$61B$	$245\mu s$	$4.3ms$
Hamming(12, 8)	$1/12$	$81B$	$391\mu s$	$3.9ms$
Hamming(16, 11)	$1/16$	$83B$	$3ms$	$3.9ms$

V. Test FEC Codes on ZigBee Device

Due to the limited storage resource and low computation ability of ZigBee devices, different FEC coding techniques exhibit significant performance differences when being applied on ZigBee, including encoding time and decoding time, which can highly influence the link throughput of ZigBee. In order to select appropriate FEC coding techniques to achieve high link throughput, we test two kinds of coding techniques with low computation complexity and storage-requirement on TelosB nodes. One is $Hamming$ coding, and another is BCH coding. Since the data is encoded, the relationship between the encoded data and CRC does not follow the CRC rule. We disable the CRC filtering of the receiver and set the encoded CRC in the packet's payload.

We test time complexities and the storage resource consumptions of 3 $Hamming$ and 3 BCH coding techniques in our experiments. The results are shown in Table I. t/n in the table means tolerant errors (t bits) in each encoded group (n bits) of the coding technique. Though BCH coding techniques have higher correction ability than $Hamming$, the valid data of BCH coding techniques in each encoded group is less than that of $Hamming$ techniques. In addition, the decoding time of BCH coding techniques is far larger than that of $Hamming$ coding techniques. The encoding time of all coding techniques is much shorter than the decoding time, except for $Hamming(16, 11)$. When applying $Hamming(16, 11)$ on ZigBee nodes, the nodes cannot use the fast table look-up way to encode the data due to the high requirement of RAM resource.

VI. Real-time Adaptive Transmission with WiFi Interference

In this section, we present the proposed real-time adaptive transmission scheme for ZigBee packets to survive from WiFi interference. With the WiFi traffic model above, we can estimate the distribution of WiFi interfering time in a ZigBee packet. 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. Thus, given a FEC coding technique, the probability that the ZigBee packet fails be recovered can be estimated. In this way, we derive the transmission efficiencies of the FEC coding techniques in different WiFi traffic cases, and hence get the optimal coding strategy for ZigBee devices based on their measurement on the current channel.

A. Distribution of errors in ZigBee packet

Denote $f_a(x)$ as the probability density function (pdf) for the arrival process of WiFi frame clusters, and $f_c(x)$ as the pdf for the size of WiFi frame clusters. It has been validated [3] that the arrival process of WiFi frame clusters has the distinct feature of self-similarity and can be accurately characterized by Pareto model (though it is claimed for *white space*). Leveraging the Pareto model, we give $f_a(x)$ as follows:

$$f_a(x) = \frac{\lambda_a}{\lambda - 1}\left(\frac{1}{x}\right)^{\frac{2\lambda_a - 1}{\lambda_a - 1}}, x > 1 \tag{2}$$

where λ_a is the average inter-arrival time of WiFi frame clusters.

From Equation 1, $f_c(x)$ can be given as:

$$f_c(x) = \frac{\lambda_c}{\lambda - 0.2}\left(\frac{1}{x + 0.8}\right)^{\frac{2\lambda_c - 0.2}{\lambda_c - 0.2}}, x > 0.2 \tag{3}$$

The arrival time of k WiFi frame clusters is:

$$T(k) = \sum_{j=1}^{k} t_j \tag{4}$$

where t_j is the arrival time of jth WiFi frame clusters.

Hence, the pdf of $T(k)$ is:

$$f_a^k(x) = f_a(x) \otimes f_a(x) \otimes \ldots \otimes f_a(x) \tag{5}$$

where $k = 1, 2, 3, 4,\ f_a^1(x) = f_a(x)$.

Thus, the probability that i WiFi frame clusters are within the air-in time of a ZigBee packet is:

$$P\{n = i\} = \int_0^T f_a^i(x)\mathrm{d}x \int_T^\infty f_a^{i+1}(x)\mathrm{d}x \tag{6}$$

where $1 \le i \le 3$.

Denote Z as the sum of the sizes of WiFi frame clusters within the air-in time of ZigBee packet. We suppose the arrival time of the WiFi clusters is independent of the size. Hence, the pdf of Z is:

$$f_Z(x) = \sum_{i=1}^{3} f_c^i(x) P\{n = i\} \tag{7}$$

where $f_c^k(x) = f_c(x) \otimes f_c(x) \otimes \ldots \otimes f_c(x)$, $k = 1, 2, 3$.

We regard Z as the interfering time in a ZigBee packet. Actually, the interfering time is overestimated, as there are many small idle intervals within the WiFi frame clusters.

The number of error bits in the interfering duration depends on the current SNIR. As WiFi interference does not always cause errors in ZigBee packet, the errors may be fragmentary and spread in the interfering duration. Denote p_{eI} as the bit error rate (BER) in ZigBee packet when there is WiFi interference. p_{eI} changes with the SNIR in the channel. Since the physical layer of ZigBee uses O-QPSK modulation with half-sine pulse shaping [22], p_{eI} can be calculated as follows.

$$\begin{aligned} p_{eI} &= Q(\sqrt{2\gamma * SNIR(t)}) \\ SNIR(t) &= 10 \lg \frac{S(t)}{I(t) + \sigma_n^2} + 9dB \end{aligned} \tag{8}$$

where $SNIR(t)$ is the signal-interference-noise-ratio of ZigBee interfered by WiFi interference and noise. $S(t)$ is the strength of ZigBee signal, which can be obtained by ZigBee receiver node when the node detects a ZigBee packet. $I(t)$ is the strength of WiFi interference. σ_n^2 is the average strength of noise that can be measured when the channel is idle. The processing gain (PG) for ZigBee signal detection is $9dB$ and $\gamma \approx 0.85$.

We ignore the errors caused by noise when there is no WiFi interference. Hence, during Z interfering time, the number of error bits in a ZigBee packet is:

$$N_Z = \sum_{j=1}^{\lceil Z \cdot R \rceil} \Theta_j \tag{9}$$

where, $R = 250kbps$, and

$$\Theta_j = \begin{cases} 1, & p_{eI} \\ 0, & 1 - p_{eI} \end{cases} \tag{10}$$

B. Adaptive encoding strategy

To apply FEC coding method into the data in ZigBee packet, the data needs to be encoded group by group as described above. If the number of errors within an encoded group exceeds the error-correcting code's capability, it fails to recover the original data. An encoded ZigBee packet can be successfully delivered only if each encoded group in the packet can recover the data, otherwise the data cannot pass the CRC check.

However, error bits in ZigBee packet due to WiFi interference does not spread evenly, which leads to failures of some groups in the packet and as a result the whole packet fails. To ameliorate the problem, two approaches are employed in ZigBee packet in the proposed scheme. One is bit interleaving, and another is *segment-based CRC*.

1) Bit interleaving: Bit interleaving spreads bursting errors to the whole packet, thus fully utilizing the correction ability of all the groups in the packet. According to the size of the encoded group, e.g., n-bits, all L-byte encoded data in the packet will be treated as $n \times \lfloor 8L/n \rfloor)$ sequence of bits. Bit interleaving spreads each bit in a group in the whole packet and separate them by $\lfloor 8L/n \rfloor)$ bits. The receiver reverses the interleaving after receiving the packet. Thus, the order of bits in each group is recovered while some bursting errors in the packet are spread in different groups in the packet.

Suppose the errors are uniformly spread in the packet after the bit interleaving. Then, the probability that each encoded group can recover the data therein is $P\{N_Z/W < t\}$, where W is the number of encoded groups in ZigBee packet. For convenience of calculation, we approximately regard Z interfering time will cause $Z \cdot R \cdot p_{eI}$ error bits. Hence, the probability of each encoded group in the packet can recover the data is:

$$p = P\{\frac{Z \cdot R \cdot p_{eI}}{W} < t\} = \int_0^{\frac{t \cdot W}{R \cdot p_{eI}}} f_Z(x)\mathrm{d}x \tag{11}$$

2) Segment-based CRC *mechanism:* To survive the corrupted ZigBee packet in which only part of encoded groups fails, we design *segment-based CRC* mechanism. The basic idea is that we partition unencoded data in ZigBee packet into several segments. For each segment, a 2-bytes CRC is set. Hence, when some data in one segment fail to be recovered, the segment can be detected and the receiver can keep those data in correct segments. However, multiple *segment-based CRCs* set in the packet incur extra overhead. The optimal number of *segment-based CRC* is computed as follows.

Suppose the L-bytes data in ZigBee packet are partitioned into M segments. Each segment contains $\frac{8(L-2M)}{Mn}$ encoded groups. Each encoded group contains k-bits raw data. Hence, the expectation of the number of correct raw data can be obtained by receiver is:

$$N(M) = p_h \sum_{j=0}^{M} jk \cdot \frac{8(L-2M)}{Mn} \cdot C_m^j p^{j \cdot \frac{8(L-2M)}{Mn}} (1 - p^{\frac{8(L-2M)}{Mn}})^{M-j} \tag{12}$$

where p_h is the probability that the packet header survives from the interference. The optimal number of *segment-based CRC* M^* is:

$$M^* = \arg\max_M N(M) \tag{13}$$

Then, we can derive the transmission efficiency of ZigBee packet using one of the six tested coding techniques is:

$$\eta = \frac{N(M^*)}{T_e + T_d + T_t + T_a + T_\iota} \tag{14}$$

where T_e is the encoding time, T_d is the decoding time, T_t is the air-in time of packet ($4ms$), T_a is the air-in time of ACK ($352\mu s$) and T_ι is the TX/RX state switch time ($192\mu s$).

Hence, with Equations above, we can obtain the optimal coding technique among the six candidates when given λ_a, λ_c and SNIR, as well as the optimal number of *segment-based CRC*. The calculation is complex. We use PC to conduct the calculation and condense the results with several simple encoding rules that can be easily applied in resource-limited ZigBee devices, as shown in Table II.

Note that, we give the rules only in the condition of $\lambda_a < 3.2$, because this condition has covered most cases when WiFi interferers are active according to the data traces. The "active" here means the channel utilization ratio keeps being over 0.005 for over $100ms$. We observe that the case when the utilization ratio is below 0.005 usually lasts for several seconds with long-term *white space* and the arrival process model in [3] does not fit now.

C. Implementation of real-time adaptive transmissions scheme

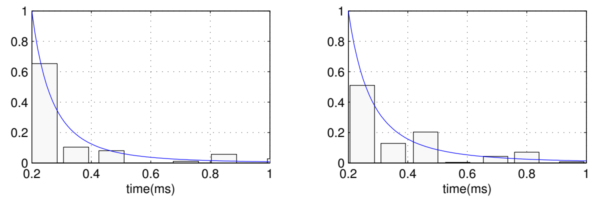
To implement the real-time adaptive transmission on ZigBee device, the key issue for ZigBee sender is obtaining the real-time λ_a, λ_c and RSSI of the WiFi channel. With these three parameters, the sender can choose the appropriate coding technique in Table II.

1) Modeling the interference with partial channel sampling: Since the arrival time of WiFi frame clusters and their sizes can be characterized only within $100ms$ window as analyzed above, the ZigBee sender needs to frequently sample the channel to update the models. However, due to the half-duplex communication mode, ZigBee device cannot sample the channel while it is transmitting a packet. Considering the fact of distinct gap between encoding time and decoding time of ZigBee device, ZigBee sender has to wait for the receiver to finish packet decoding before sending the next packet. We employ the waiting time for channel sampling and modelling. Specifically, the sender sets a moving window with size $100ms$ for collected samples in waiting time, derives the models and decides the encoding technique for the next packet according to the encoding rules in Table II, thus achieving real-time adaptive transmissions.

To evaluate the accuracy of models established by samples in the disjunctive waiting time, we conduct analysis on the data trace in VWAVE 2009. For each window in the data trace, samples are taken in every other $5ms$ and estimate the values of λ_a, λ_c. Using the values, we estimate the arrival

TABLE II: Adaptive encoding rules $\lambda_a < 3.2$

<table>
<tr><th>λ_c/λ_a</th><th>no need coding</th><th>Hamming(12, 8)
3 segment-based CRC</th><th>Hamming(7, 4)
6 segment-based CRC</th><th>BCH(15, 5)
12 segment-based CRC</th><th>BCH(15, 1)
12 segment-based CRC</th></tr>
<tr><td>0.1</td><td>$SNIR \geq -4dB$</td><td>$-8dB \leq SNIR < -4dB$</td><td>$-10dB \leq SNIR < -8dB$</td><td>$SNIR < -10dB$</td><td>N/A</td></tr>
<tr><td>0.2</td><td rowspan="2">$SNIR \geq -3dB$</td><td>$-7dB \leq SNIR < -3dB$</td><td>$-12dB \leq SNIR < -7dB$</td><td>$-23dB \leq SNIR < -12dB$</td><td>$SNIR < -23dB$</td></tr>
<tr><td>0.3</td><td>$-6dB \leq SNIR < -3dB$</td><td>$-9dB \leq SNIR < -6dB$</td><td>$-15dB \leq SNIR < -9dB$</td><td>$SNIR < -15dB$</td></tr>
<tr><td>0.4</td><td rowspan="6">$SNIR \geq -2dB$</td><td rowspan="3">$-5dB \leq SNIR < -2dB$</td><td>$-9dB \leq SNIR < -5dB$</td><td>$-13dB \leq SNIR < -9dB$</td><td>$SNIR < -13dB$</td></tr>
<tr><td>0.5</td><td rowspan="2">$-8dB \leq SNIR < -5dB$</td><td>$-11dB \leq SNIR < -8dB$</td><td>$SNIR < -11dB$</td></tr>
<tr><td>0.6</td><td>$-10dB \leq SNIR < -8dB$</td><td>$SNIR < -10dB$</td></tr>
<tr><td>0.7</td><td rowspan="3">$-4dB \leq SNIR < -2dB$</td><td rowspan="2">$-7dB \leq SNIR < -4dB$</td><td rowspan="2">$-9dB \leq SNIR < -7dB$</td><td rowspan="3">$SNIR < -9dB$</td></tr>
<tr><td>0.8</td></tr>
<tr><td>0.9</td><td>$-6dB \leq SNIR < -4dB$</td><td>$-9dB \leq SNIR < -4dB$</td></tr>
</table>



(a) $\lambda_c = 0.4$ (b) $\lambda_c = 0.5$

Fig. 4: Comparison between the estimation and real results of the WiFi cluster sizes in the next $5ms$.

time of WiFi frame clusters and their sizes in the next $5ms$ and compare the estimation result with the true samples in that time.

Fig. 4 shows the comparisons between the estimation result based on the partial sampling and the statistical results of real WiFi cluster sizes with two typical examples $\lambda_c = 0.4$ and $\lambda_c = 0.5$, where the curves are the pdf of WiFi frame cluster sizes estimated by the power law model with the partial channel sampling data. It can be seen, the estimation result is much close to the statistical real result, showing the effectiveness of the model with partial channel sampling for cluster size. Due to the space limit, we do not give the comparison about the arrival process of WiFi frame clusters, as it has been validated in [3] that the arrival time of clusters can be accurately modeled with even a very low sampling frequency.

2) Sampling the channel with the interrupt mechanism: Note that, during the channel sampling, ZigBee device does not need to keep sampling the channel. An interrupt mechanism is usually built in the ZigBee motes. Whenever the CCA pin state changes, it triggers an interrupt to the micro-controller. Then, the controller checks the SFD pin to judge whether there is a decodable ZigBee packet arriving or not. In this way, ZigBee device can record the samples of WiFi frame clusters and the arrival times.

The parameter of SNIR is measured at the receiver side. ZigBee receiver measures the RSSI of ZigBee signal, WiFi interference and noise independently. When both CCA and SFD pins trigger interrupts, the current RSSI can be recorded as the power of ZigBee signal S. When only CCA pin trigger interrupt with going low, the current RSSI can be recorded as the power of interference I. Considering that RSSI output by ZigBee device is conducted within $128\mu s$ which is relative large to the air-in time of WiFi packets, the time for the RSSI sample may partially overlapped by WiFi signal, leading to variety of sequential RSSI samples for WiFi interference strength. We choose the maximum value of RSSI for WiFi interference within $50ms$ as the power of interference I, as the strength of WiFi signal can be regarded as a constant within the coherence time (about $50ms$ when the velocity of any movement in the scenario is less than $1m/s$) due to the temporal channel correlation [22]. As for the power of noise, it can be measured with averaging the RSSI samples when the channel is idle.

After the receiver obtains the current $SNIR(t)$, it sends the information to the sender in the ACK packets. Along with the information, the receiver also needs to add the IDs of failed segments in last packet into the ACK, so that the sender get to know the data in which segment should be retransmitted. According to Table II, the overhead due to denoting the IDs of failed segment is small, as there are no more than 12 segments in the packet.

VII. Experimental Results

In this section, we present the evaluation results of the real-time adaptive transmission scheme. The scheme is implemented on TelosB motes equipped with 802.15.4 radios. A laptop (called laptop A) equipped with 802.11 compliant Intel PRO 3945ABG NICs is used as WiFi interferes. We use D-ITG [23], a high-fidelity Internet traffic generator, to generate the real-life WiFi traffic at different rates. It has been show that D-ITG can reproduce realistic traffic patterns under a wide range of network settings [23].

We compare the proposed scheme to two baseline protocols. One is WISE protocol proposed in [3], and another is $BuzzBuzz$ protocol in [4] (suppose a fixed $Hamming(12, 8)$ coding technique is used). These two protocols are specially designed for ZigBee devices under the interference of WiFi. For fair comparison, we run these three approaches on three pairs of TelosB motes simultaneously in the same WiFi interference environment. Specially, we set the frequency channel of the motes for WISE, the proposed scheme and BuzzBuzz at 802.15.4 channel 11, channel 12 and channel 13, respectively, and set that of WiFi interferer at 802.11g channel 1. In this way, the three TelosB motes will not interfere with each other, while the WiFi signal can interfere the three motes due to the wide bandwidth of WiFi signal. Note that, WiFi transmitters do not emit uniform power across their bandwidth. The power density at the center of the bandwidth is generally higher than that at the border. Therefore, the interference strength in 802.15.4 channel 12 (i.e., the channel for our proposed scheme) which is near to the center of 802.11g channel 1, is no less than those in other two 802.15.4 channels. In addition, we connect the receiver of each pair of TelosB motes directly to another laptop (called laptop B) via three USB interfaces, and put laptop B $20m$ away from laptop A. For the senders, we bind them together and place them at different places with distance $5m$, $10m$, and $15m$ to laptop B, respectively.

We suppose the three TelosB sender motes always have packets to delivery, thus testing the ZigBee link capacity at different WiFi traffic rates. This test is mainly for evaluating the performance of data-flow ZigBee applications. The three TelosB receiver motes record the number of data bytes successfully received every $100ms$ and send the number to laptop B via the USB interfaces. In this way, the link throughput of the three pairs of motes is recorded. In addition, the receiver motes also record the average delivery delay of packets every $100ms$ and send the delay value to laptop B. Meanwhile, we do not count the retransmission time into the delay for the proposed scheme as long as there is at least one segment of the packet is successfully received. This is because the average packet delivery delay is mainly for evaluating the performance of command-delivery ZigBee applications where the size of each message is usually small.

We compare the performance of the proposed scheme to that of WISE and BuzzBuzz under different levels of WiFi interference, including the throughput and $SNIR$. For each case, we made experiment for $5mins$.

Fig. 5 shows the throughput of $WISE$, $BuzzBuzz$ and the proposed real-time adaptive transmission scheme (denoted by RAT in the figure). When WiFi throughput and the distance between ZigBee motes increase, ZigBee throughput of three approaches drops. Meanwhile, WiFi throughput plays a major role. However, the throughput of WISE and BuzzBuzz drops more quickly than that of RAT. In particular, when WiFi throughput increases to $8Mbps$, the ZigBee throughput of both $WISE$ and $BuzzBuzz$ drops near to zero, while that of RAT still keeps around $10kbps$. The uneven drop of the throughput is due to the discrete correction capability of the adaptive encoding techniques.

Fig. 6 shows the average packet delivery delay when using $WISE$, $BuzzBuzz$ and RAT. With the increment of WiFi throughput and the distance between ZigBee motes, the average packet delay of $WISE$ and $BuzzBuzz$ increases sharply, while that of RAT rises a little. When WiFi throughput increases to $6Mbps$, the average packet delay of $WISE$ and $BuzzBuzz$ exceeds $1s$. However, the delay of RAT is no more than $150ms$ even when WiFi throughput increases to $10Mbps$. The low average packet delivery delay is of great significance to the delay-sensitive ZigBee applications, such as wireless game controlling.

Note that, as we use D-ITG to to generate the real-life WiFi traffic, the WiFi data rate actually varies during the experiments for a given WiFi throughput in D-ITG. Also, both sizes of WiFi frame clusters and the arrival time keep changing in the WiFi traffic.

Finally, we give a discussion on the energy consumption of ZigBee nodes with the three schemes. We are mainly concerned with the energy consumption due to communication activities (e.g., listening, receiving and transmitting) of nodes, as these activities consume most of the energy of the nodes. During the communication of ZigBee nodes, the three schemes all require the nodes to keep listening to (or sampling) the channel, receiving or transmitting. That is to say, the time during which the node' radio module keeps active, can be a metric for the energy consumption. Since RAT has the prior performance in throughput and packet delivery delay, the time needed by RAT to complete a given transmission task (e.g., data flow traffic or command delivery traffic) will be shorter than those of others in WiFi interfering environment, i.e., RAT consumes less energy.

VIII. Conclusions

In this paper, we propose a real-time transmission scheme that facilitates ZigBee links to achieve high throughput and low delivery delay in WiFi interference environment. We employ adaptive encoding idea to recover the corrupted ZigBee packets according to the recent WiFi interference. In order to choose appropriate coding technique, we predict the error distribution in ZigBee packet with the observation that WiFi frame clusters well fit the power law distribution. Hence, with testing the performance of a series of light-weight FEC coding techniques on ZigBee devices, we deduce the optimal coding strategy for ZigBee in different WiFi interfering environment and build in an optimal *segment-based CRC* mechanism in it to reduce the packet failure probability. We validate the interference prediction on the open data trace with partial channel sampling. Extensive experiments are made on a testbed of 802.11 netbook and three pairs of $802.15.4$ TelosB motes. The results show the TelosB mote with the proposed scheme can well adapt its transmissions with the external WiFi interference and the performance remarkably outperforms that of $WISE$ and $BuzzBuzz$. In particular, when WiFi throughput is over $8Mbps$, the ZigBee throughput with the proposed scheme can be about $10kbps$ and the average packet delivery delay is no more than $150ms$, while those of existing works is near to $0kbps$ and over $1s$, respectively. The performance achieved by the proposed scheme can be of great significance to the performance-sensitive ZigBee applications in practice.

Acknowledgement

The work presented in this paper was supported in part by the NSF of China with Grant 61272053 and Hong Kong Scholars Program.

References

[1] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan, *Understanding and mitigating the impact of RF interference on 802.11 networks*, In ACM SigCOMM, 2007.

[2] R. Chandra, R. Mahajan, V. Padmanabhan, and M. Zhang, *Crawdad data set microsoft/osdi2006 (v. 2007-05-23)*, 2007.

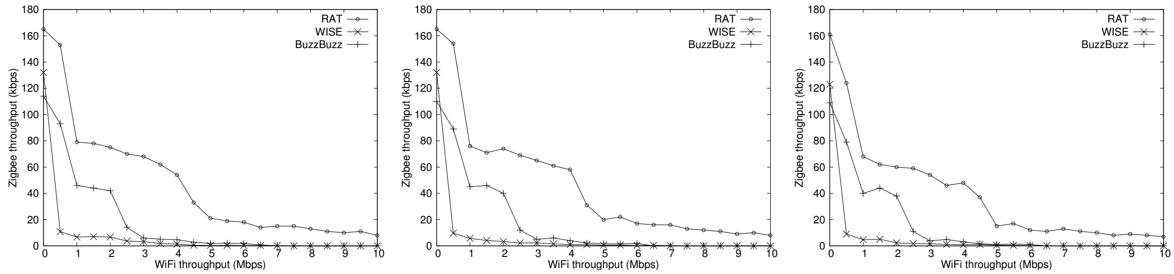
[3] Jun Huang, Guoliang Xing, Gang Zhou, and Ruogu Zhou, *Beyond Co-existence: Exploiting WiFi White Space for ZigBee Performance Assurance*, In ICNP, 2010.

[4] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, Andreas Terzis, *Surviving Wi-Fi Interference in Low Power ZigBee Networks*, SenSys, 2010.

[5] Axel S., Voicu FG., *Coexistence of ZigBee with other Systems in the 2.4 GHz-ISM-Band*, In IEEE Instrumentation and Measurement Technology, 2005 May, Ottawa, Canada, pp. 1786-1791.

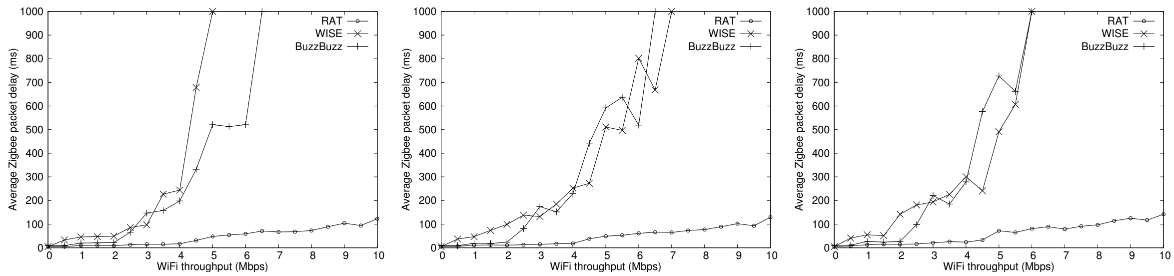
[6] Soo YS, Hong SP, Wook HK, *Mutual interference analysis of ZigBee and WiFi-b*, Computer and Telecommunications Networking, no. 51, 2007, pp. 3338C3353.

[7] S. Pollin, I. Tan, B. Hodge, C. Chun, and A. Bahai, *Harmful Coexistence Between 802.15.4 and 802.11: A Measurement-based Study*, In Proc. of CrownCom, 2008.



(a) distance=5m (b) distance=10m (c) distance=15m

Fig. 5: ZigBee throughput vs WiFi throughput.



(a) distance=5m (b) distance=10m (c) distance=15m

Fig. 6: ZigBee packet delivery delay vs WiFi throughput.

[8] J.-H. Hauer, V. Handziski, and A. Wolisz, *Experimental Study of the Impact of WLAN Interference on IEEE 802.15.4 Body Area Networks*, In EWSN, 2009.

[9] *Coexistence of Wireless Personal Area Networks With Other Wireless Devices Operating in Unlicensed Frequency Bands*, IEEE Std 802.15.2, 2003.

[10] C. Won, J.-H. Youn, H. Ali, H. Sharif, and J. Deogun, *Adaptive Radio Channel Allocation for Supporting Coexistence of 802.15.4 and 802.11b*, In VTC, 2005.

[11] J. Mitola and G. M. Jr., *Cognitive radio: making software radios more personal*, IEEE Personal Communications, vol. 6, no. 4, pp. 13-18, Aug. 1999.

[12] Q. Zhao, L. Tong, A. Swami, and Y. Chen, *Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework*, IEEE Journal on Selected Areas in Communications, vol. 25, no. 3, pp. 589-600, 2007.

[13] I. A. Akbar and W. H. Tranter, *Dynamic Spectrum Allocation in Cognitive Radio Using Hidden Markov Models: Poisson Distributed Case*, in Proceedings of IEEE SoutheastCon, pp. 196-201, 2007.

[14] R. Gummadi, H. Balakrishnan, and S. Seshan, *Metronome: Coordinating Spectrum Sharing in Heterogeneous Wireless Networks*, In First International Workshop on Communication Systems and Networks (COMSNETS), 2009.

[15] J. Hou, B. Chang, D.-K. Cho, and M. Gerla, *Minimizing 802.11 Interference on ZigBee Medical Sensors*, In BodyNets, 2009.

[16] Xinyu Zhang and Kang G. Shin, *Enabling Coexistence of Heterogeneous Wireless Systems: Case for ZigBee and WiFi*, MOBIHOC, 2011.

[17] Cioni, S., De Gaudenzi, R., Rinaldo, R., *Adaptive coding and modulation for the reverse link of broadband satellite networks*, in IEEE Global Telecommunication Conference, USA, Dec. 2004, pp. 1101-1105.

[18] Anas Basalamah, Hiroki Sugimoto and Takuro Sato, *Rate Adaptive Reliable Multicast MAC Protocol for WLANs*, IEEE Vehicular Technology Conference, May, 2006, pp. 1216-1220.

[19] C. Noda, S. Prabh, M. Alves, T. Voigt, C. A. Boano, *Crawdad data set cister/rssi (v. 2012-05-17)*, 2012.

[20] Caleb Phillips, Suresh Singh, *Crawdad data set pdx/vwave (v. 2009-07-04)*, 2009.

[21] A. Solanas, R. Manolov1, and V. Sierra, *Lag-one autocorrelation in short series: Estimation and hypotheses testing*, International Journal of Methodology and Experimental Psychology, vol. 31, no. 2, 2010, pp. 357-381.

[22] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ: Prentice Hall, 1996.

[23] A. Dainotti, A. Botta, A. Pescap. A tool for the generation of realistic network workload for emerging networking scenarios. Computer Networks (Elsevier), 2012, vol. 56(15), pp 3531-3547.