Smoggy-Link: Fingerprinting Interference for Predictable Wireless Concurrency

Meng Jin[1], Yuan He[2], Xiaolong Zheng[2], Dingyi Fang[1], Dan Xu[1], Tianzhang Xing[1], Xiaojiang Chen[1]
[1]Northwest University, P.R. China
[2]Tsinghua University, P.R. China
mengj@stumail.nwu.edu.cn, {he, xiaolong}@greenorbs.com, {dyf, xudan, xtz, xjchen}@nwu.edu.cn

***Abstract*—Operating in unlicensed ISM bands, ZigBee devices often yield poor throughput and packet reception ratio due to the interference from ever increasing wireless devices in 2.4 GHz band. Although there have been many efforts made for interference avoidance, they come at the cost of miscellaneous overhead, which oppositely hurts channel utilization. Our empirical results show that, a specific interference is likely to have different influence on different outbound links of a ZigBee sender, which indicates the chance of *concurrent transmissions*. Based on this insight, we propose Smoggy-Link, a practical protocol to exploit the potential concurrency for adaptive ZigBee transmissions under harsh interference. Smoggy-Link maintains an accurate link model to describe and trace the relationship between interference and link quality of the sender's outbound links. With such a link model, Smoggy-Link can obtain fine-grained spatiotemporal link information through a low-cost interference identification method. The link information is further utilized for adaptive link selection and intelligent transmission schedule. We implement and evaluate a prototype of our approach with TinyOS and TelosB motes. The evaluation results show that Smoggy-Link has consistent improvements in both throughput and packet reception ratio under interference from various interferer.**

I. Introduction

The explosive growth of wireless and mobile devices boosts proliferation of heterogeneous network technologies on the 2.4 GHz ISM (Industrial Scientific Medical) band. Typical examples include Wi-Fi, ZigBee, Bluetooth, 2.4 GHz cordless phones, surveillance cameras, game controllers, and 2.4 GHz RFID. While those technologies bring to people a lot of convenience and efficiency, a serious problem attracts increasing attention: spectrum sharing among incompatible wireless technologies has led to a severe cross-technology interference problem [1–5], especially for the low-power technologies such as ZigBee [4, 5].

The existing approaches of interference resolution largely aim at communicating over non-overlapping segments of the spectrum [6–8]. The ISM band is becoming increasingly crowded, making it difficult to find an interference-free channel. This led the researchers to focus on developing interference avoidance solution in time domain [4, 9, 10]. These time domain solutions usually come at the cost of sacrificing the efficiency of channel utilization for conservative backoff, which limits network throughput. Clearly, there is an inherent conflict between interference avoidance and channel utilization. This motivates us to reconsider the way to handle interference from a new aspect.

Instead of merely avoiding the interference, we argue that there actually exists abundant opportunity to transmit packets *concurrently* with the interference. Our key observation in Section III shows that a specific interference is likely to have different influence on different outbound links of a ZigBee sender. Even under strong interference, such as Wi-Fi, there is still certain chance for any one of the receivers to successfully decode the sender's packet. This is due to implementations and behaviors of wireless communication techniques such as DSSS (direct-sequence spread spectrum) modulation scheme and exposed terminal phenomenon. Exploiting such opportunities can significantly improve network throughput.

However, we may meet two critical challenges towards the above goal: first, considering the dynamic network environment, the feasible concurrency pattern (which link can concurrently transmit) usually seems unpredictable and changeable, thus we need an accurate link model to describe and trace the relationship between the interference and the quality of each outbound link. Second, to avoid the collision between the ACK and the interference, which would undermine link estimation, we need to carefully schedule transmission of ACKs to make it arrive at the sender during the idle space between interference frame clusters.

In this paper, we propose Smoggy-Link, a practical protocol to exploit the potential concurrency for adaptive transmission under interference. The idea of Smoggy-Link is based on our observations that *link quality is highly related to interference*. Therefore, given the feasibility of interference identification according to the featured patterns of interference signals, we can utilize the low-cost interference information, to obtain fine-grained spatiotemporal link information. The link information can be further utilized for adaptive link selection for concurrent transmissions. In addition, we also observe *predictable patterns* of data arrival process of interferences. This enables a node to predict the idle space of the interference and intelligently schedule the transmission of data and ACKs to achieve both high channel utilization and low ACK collision probability. The contributions of this paper are summarized as follows:

- Based on the observation of the abundant opportunity for concurrent transmissions, we propose Smoggy-Link, an adaptive transmission protocol that can fully exploit concurrency to maximize the network throughput while achieving expected packet reception ratio.

978-1-5090-3281-5/16/$31.00 ©2016 IEEE

- We present a novel link model to accurately characterize the relationship between link quality and the interference. The model is utilized to estimate the link quality and idle space distribution under different interference in realtime.
- We implement and evaluate a prototype of our approach with TinyOS and TelosB motes. The evaluation results show that Smoggy-Link has consistent improvements in both throughput and packet reception ratio under interference from various interferers.

The rest of the paper is organized as follows. Section II summarizes the related work. In Section III we present our empirical measurement study that motivates this work. The design of Smoggy-Link is given in Sections V, VI and VII. In Section VIII we evaluate Smoggy-Link's performance through extensive experiments. Section IX concludes the paper.

II. Related Work

A. Wireless interference

Wireless links are unreliable and prone to losses due to noise and interference [4, 11–17], and thus wireless interference has been the topic of much recent researches. Work in this area falls under two broad categories:

Interference avoidance. To avoid the interference, early MAC layer protocols, such as B-MAC [12] and X-MAC [13] will conduct CCA before each transmission and perform exponential backoff if the channel is busy. However, since the potential interference is fundamentally uncertain and difficult to be accurately predicted, size of the backoff window is usually conservatively chosen. Thus a lot of idle slots are left unused in the channel. To further improve channel utilization, the authors of [4] propose an approach to enable ZigBee to transmit packets during the white space of the interference by learning transmission characteristics of the interferences. However, its theoretical upper bound is still m time slots to transmit m wireless packets. Smoggy-Link, on the other hand, enables different senders to transmit concurrently without interfering with each other.

Interference tolerance. Different from interference avoidance, the idea of interference tolerance enables the sender to transmit concurrently with the interference. For example, protocols like Flash Flooding [18], Chorus [19] and Glossy [20] attempt to exploit capture effect for concurrent data transmission. However, they can be only applied in flooding or broadcasting scenarios, where transmitted packets must carry the same data. Based on the understanding of basic timing and requirements of concurrency, CoCo [21] advocates simultaneous accesses from multiple senders to a shared channel. However, CoCo rely on control packets to adjust transmission probability of different senders, which limits its application in cross-technology concurrency. Unlike these schemes, however, Smoggy-Link delivers a transmission system that enables concurrency with multiple different wireless technologies.

B. Interference Awareness

It has been shown that accurate interference identification is feasible and profitable for interference resolving approaches.

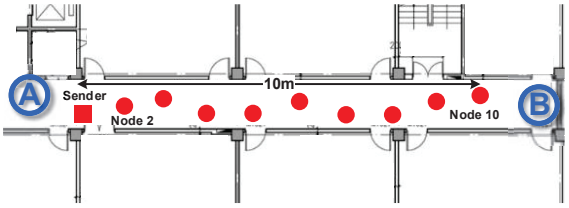


Fig. 1: Experiment Setup.

For example, ZiSense [22] utilize the featured patterns of the interferences to avoid the unnecessary wake-ups. CrossZig [14] features physical layer hints to infer and recognize interference patterns and harnesses this to adapt the recovery mechanism. SoNIC [23] proposes a method to classify non-ZigBee interferences which helps to identify the corrupted bits.

Our work aims at bridging the above two directions. Based on interference identification, we can obtain fine-grained spatiotemporal link information which helps to design more efficient concurrency approach.

III. Impact of Cross-Technology Interference on ZigBee

To better understand how the cross-technology interference impacts on ZigBee, we conduct empirical studies under different interferences. We focus on three technologies that are prevalent in today's network environments: Bluetooth, Wi-Fi and Microwave.

A. Experimental Setup

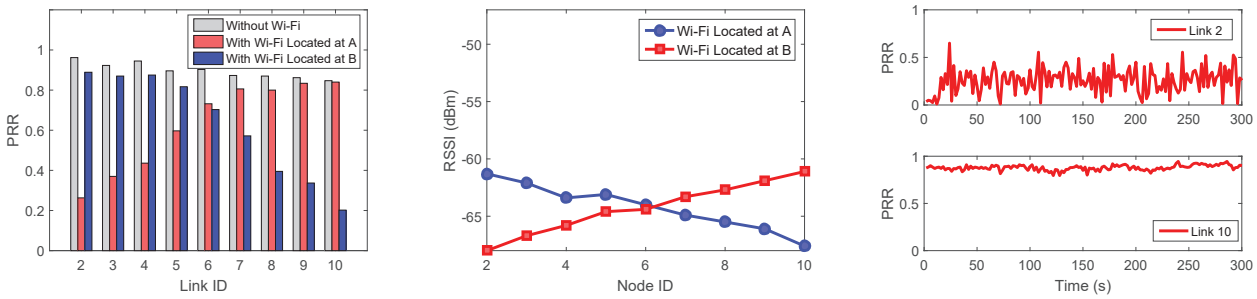
We deployed ten TelosB nodes in a corridor in a university building. The deployment scenario is depicted in Figure 1. Node 1 acts as a sender and Node 2-10 act as the receivers. The links between the sender and the receivers are denoted as Link 2-10. The transmission power of the sender is tuned to level 5. The experiments are conducted in the 12th channel.

In the experiment, the sender periodically broadcasts 54-byte payload packets at a rate of 100 packets per second, during which the receivers listen and record the sequence numbers of the received packets. In each run, we place one interferer (Wi-Fi, Bluetooth, or microwave oven) at one of the marked positions (A or B) in Figure 1. In each location, the interferer would be turned ON and generate interference signal for a period of 5 min. We compute the observed Packet Reception Ratio (PRR) first when the interferer is turned OFF and next when it is ON.

B. Impact of Wi-Fi

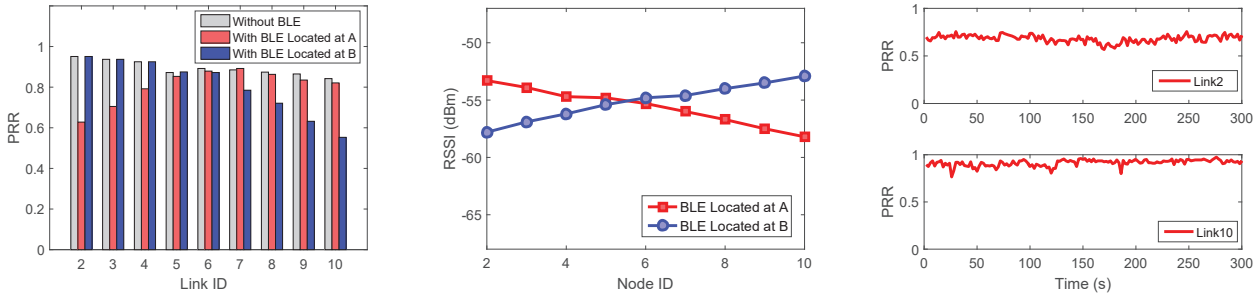
We create Wi-Fi interference using an iPhone 6S smartphone as AP and another iPhone 6S smartphone as clint. The data rate is not constant when the interference is ON and it ranges from $2Mbps \sim 8Mbps$. In each experiment, we fix the locations of the ZigBee nodes and place both the Wi-Fi AP and Clint at location A or B, 50 cm away from each other.

Figure 2(a) shows the PRR of different ZigBee links with and without interference from Wi-Fi. We can see from the



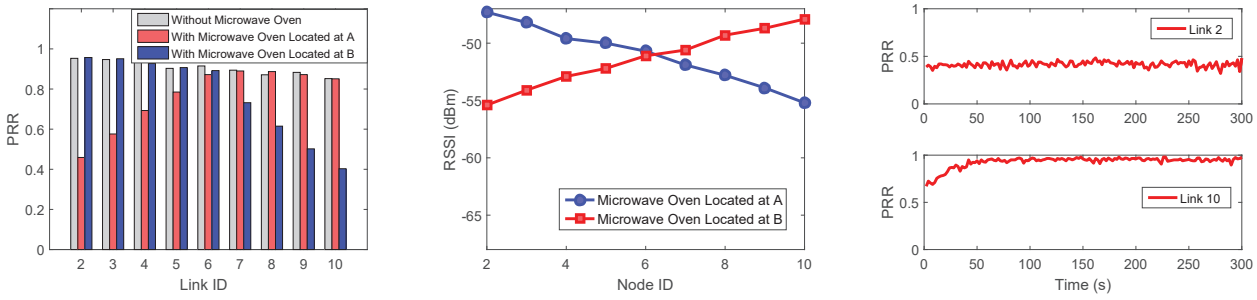
(a) Average Packet Reception Ratio (b) RSSI (c) Packet Reception Ratio

Fig. 2: Impact of Wi-Fi.



(a) Average Packet Reception Ratio (b) RSSI (c) Packet Reception Ratio

Fig. 3: Impact of Bluetooth.



(a) Average Packet Reception Ratio (b) RSSI (c) Packet Reception Ratio

Fig. 4: Impact of Microwave.

figure that the Wi-Fi interference is likely to have different impact on different links, thus there is strong possibility for any one of the receivers to successfully decode the sender's packet even under an ongoing Wi-Fi interference. For example, when the interferer is located at A, the PRRs of the Links 2-4 is less than 50%, while that of Links 8-10 exceed 80%, indicating that the sender can transmit packets concurrently with the interference via Links 8-10. Figure 2(b) gives an explanation of the above phenomenon: the receivers located far away from the interference (such as Receivers 8-10) suffer the interference with lower RSSI than that located close to the interference (such as the Receivers 2-4).

In addition, Figure 2(a) also shows that different interferences would have different impact on the links. Specifically, the impact from the Wi-Fi interferer located at A (denoted as Wi-Fi A) is completely different from that from the Wi-Fi interferer located at B (denoted as Wi-Fi B). For example, when Wi-Fi A is ON, PRR of Node 2 is only 20.9%, while that of Node 10 reaches 86.8%. However, when Wi-Fi B is ON, the results are exactly reversed. The above observation implies that we need an interference-aware link estimator to estimate the PRRs of a sender's outbound links under different interferences. However, this requires that the PRR of a link is stable under a certain interference.

We show the impact of different interferences in time-domain in Figure 2(c). Specifically, Figure 2(c) plots the PRR traces of Link 2 under interferences from the Wi-Fi interferer located at A and B, respectively (although not shown here,

these results generalize for other links as well). The figure tells that although the link under different interferers exhibit distinct PRRs, the link quality is stable under a certain interference, especially when the PPR is high. This demonstrates the feasibility of link estimation based on the interference information.

C. Impact of Bluetooth

Then we evaluate the interference generated by Bluetooth. Bluetooth uses frequency hopping across a 79 MHz band in the 2.402-2.480 GHz range, occupying 1 MHz at any point in time. We generate the Bluetooth interference by transferring a large file between two iPhone 6S smartphones with a data rate of $1 \sim 2Mbps$.

We plot in Figure 3(a) the PRR obtained by different links, in the presence and absence of the Bluetooth interference. Compared with Wi-Fi, the Bluetooth causes less impact on link quality although it has higher RSSI as shown in Figures 3(a) and 3(b). This is because the Bluetooth exhibits a lower occupancy level than Wi-Fi due to the frequency hopping technology. In addition, Figure 3(c) shows that link quality is stable under the interference from Bluetooth, which is similar to that under Wi-Fi.

D. Impact of Microwave Oven

We use a residential microwave oven, the Galanz G90F25CN3L-C2(G2), to generate the microwave interference. We heat a litre of water in the microwave to emulate an interference typical to that emitted by these appliances. Experimental results are shown in Figure 4. As shown in the figures, except that microwave oven have less impact on link quality than Wi-Fi, the pattern of the link quality under the impact of microwave oven is similar with that under Wi-Fi and Bluetooth.

E. Summery

Based on the above experiments, the performance of ZigBee under different interferences can be summarized as follows:

- A specific interference is likely to have different impact on different links, which motivates us to mine the potential opportunity for a sender to transmit packets concurrently with the interference.
- Link quality is highly related to the interference (independent of the technology). Thus an interference-aware link estimator is badly required to select the strongest link under different interferences.

IV. Overview

Smoggy-Link is a practical protocol to exploit the potential concurrency for adaptive transmission under interference. Unlike the existing works that merely avoid the interference, we propose to mine the potential opportunity for concurrent transmission with the interference. The intuition underlying Smoggy-Link is our observation that the quality of a sender's outbound links are highly related to the ongoing interference. Thus we can obtain finegrained link information through a low-cost interference identification method, which further enables adaptive link selection and intelligent transmission schedule. Figure 5 presents the framework of Smoggy-Link.

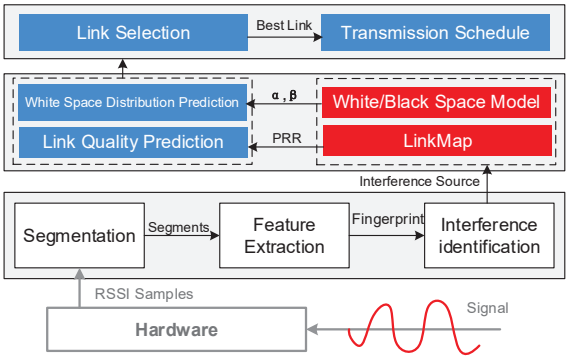


Fig. 5: Framework of Smoggy-Link.

Specifically, Smoggy-Link's design consists of the following three components:

Interference identification. The first step in the design of Smoggy-Link is to identify the interference. Indeed, almost every radio communication has a specific channel property that forms a fingerprint for that particular interference [22, 24], thus a sender can distinguish different interferences by using such a fingerprint. The targets of the interference identification component are i) creating a fingerprint for each interference in the network environment (i.e., build a fingerprint table), and ii) identifying the ongoing interference based on the obtained fingerprint table.

Interference-aware link estimation. This component acts as a bridge between the interference identification component and the subsequent data transmission component. Specifically, based on the historical link quality information under different interferences, Smoggy-Link builds a link model which describes the relationship between the interference and link quality. Then using this model, Smoggy-Link can obtain fine-grained link information only based on the low cost interference identification method. This link quality information further acts as the input of the subsequent data transmission component for further decision making.

Interference-adaptive transmission. Based on the link quality information obtained from the link estimation component, the sender adaptively selects the link with the best quality under the ongoing interference and then intelligently schedules transmission of packets. This enables the sender to maximize the throughput under the interference without sacrificing the packet reception ratio.

V. Interference Identification

As shown in [22, 24], each interference instance, independent of the technology, leaves a particular fingerprint in the channel that shapes the channel properties in a unique way. Thus a node can distinguish different interferences by using such a fingerprint measured by the built-in RSSI function of ZigBee radios. In this section, we explain the design of our interference identification method.

A. Sampling and Segmentation

Formally, the fingerprint of a certain interference can be represented in the form of a RSSI feature vector as: $F = [f^1, ..., f^K]$. As shown in [22], most interference features can be extracted in a very short time period. Thus we use a $5ms$ time window to sample the RSSI sequence in the channel, which is enough to capture the features of the interference. The sampled RSSI sequence will be fed into a segmentation module for further processing.

Target of the segmentation module is to extract the effective signal (i.e., the busy periods) from the RSSI sequence. Since an effective signals usually results in a sudden difference to the noise floor, Smoggy-Link adopts a threshold based method to detect the start and end points of each busy period. Such segmentation method is widely adopted in the existing approaches [22], and its design detail is therefore omitted here. Denote the RSSI sequence in the sampling window as: $X = \{x_1, x_2, ..., x_W\}$. The output of the segmentation method are two separated arrays: I_S (the position of the start points of the busy period) and I_E (the position of the end points of the busy period). Specifically, $I_S = [s_1, s_2, ..., s_{N_S}]$ and $I_E = [e_1, e_2, ..., e_{N_E}]$, where N_S and N_E are the number of start points and end points respectively. Without loss of generality, we assume $e_1 > s_1$ (i.e., the sampled RSSI sequence start with the busy period). Then the $k-$th busy period can be represented by $X_k^{(busy)} = [x_{I_S(k)+1}, x_{I_S(k)+2}, ..., x_{I_E(k)}]$, and the $k-$th idle period can be represented by $X_k^{(idle)} = [x_{I_E(k)+1}, x_{I_E(k)+2}, ..., x_{I_S(k+1)}]$.

B. Feature Extraction

Feature extraction module takes the set of extracted segments as input and calculates the fingerprint (i.e., the features) of the detected interference. Table I lists a set of features that are found feasible to distinguish different interferences [22, 24], which describes the *temporal behaviours* and the *energy characteristics* of the interference.

Temporal behaviours. This kind of features describe the time-domain behaviour of the interference. Particularly, due to the difference in the Mac layer parameters (such as minimum packet interval), different interferences exhibit different temporal behaviours which can be used for interference identification. We choose two features, i.e. *average on-air time* (T_{on}) and *occupancy level* (R_{occ}), to describe temporal behaviour of the interference. The description and calculation of these two features is shown in Table I (T_s is the sampling period).

Energy characteristics. Due to different PHY modulation techniques, different interferences exhibit different transmitting energy, which can be utilized for interference identification. We choose 3 features, i.e. *energy span* (E_s), *energy level* (E_l), and *energy variance* (E_v) to describe energy of the interference, and the calculation of these three features is also shown in Table I.

After extracting the above features, we can build a fingerprint (denoted as F_{det}) for the detected interference I_{det}.

C. Feature-Based Interference Identification

We exploit the *K-Means* clustering technique to discriminate different interferences based on the *cityblock* distances between their fingerprints. It is worthwhile to note that we identify the interference instance instead of the particular technology behind the interference. This is because even the interferences with the same technology may have different features (this maybe caused by their difference in hardware specifications and the location in the network environment), which makes them have different impact on the links.

Assume that there are M interferences $I = \{I_1, ..., I_M\}$ in the network environment, and their fingerprints are maintained in a fingerprint table denoted as $\mathbf{FTable} = \{F_1, ..., F_M\}$. To identify the detected interference, we first measure the cityblock distance between F_{det} and each fingerprint in $\mathbf{FTable}$, and obtains a distant set $\mathbf{D} = \{d_1, ..., d_m, ..., d_M\}$.

$$d_m = \sum_{k=1}^{N_I} |f_{det}^k - f_m^k|. \tag{1}$$

I_{det} will be identified as I_m if the distant between I_{det} and I_m (i.e., d_m) is the minimum and d_m satisfies $d_m < d_{th}$ (where d_{th} is a pre-determined threshold). In addition, to keep the $\mathbf{FTable}$ up-to-date, we use moving average to update the $\mathbf{FTable}$ as:

$$F_m = \lambda \cdot F_m + (1 - \lambda) \cdot F_{det}. \tag{2}$$

Where λ is an adjustable parameter which is set as 0.9 in our implementation. To note that, if $min(\mathbf{D}) > d_{th}$, I_{det} will be treated as a new interference. Then the node would add its feature fingerprint F_{det} into $\mathbf{FTable}$.

VI. Interference-aware Link Estimation

As shown in Section III, link quality is highly related to the interference. This means that we can predict the link quality based on the result of the interference identification component. To do so, we firstly need a model that can capture the relationship between interference and link characteristic.

In this section, we propose to build a link model to i) predict the PRR of the sender's outbound links based on the interference information; and ii) predict the arrival time of the coming white space based on the observed transmission pattern of the interference.

A. Interference-Aware Link Model

LinkMap. For computing and updating PRR, each sender S constructs a *LinkMap* to maintain PRRs of its outbound links under different interferences, forming as follows:

$$LM_{M\times N} = \begin{bmatrix} prr_{S,L_1}^{I_1} & prr_{S,L_2}^{I_1} & \cdots & prr_{S,L_N}^{I_1} \\ prr_{S,L_1}^{I_2} & prr_{S,L_2}^{I_2} & \cdots & prr_{S,L_N}^{I_2} \\ \vdots & \vdots & prr_{S,L_n}^{I_m} & \vdots \\ prr_{S,L_1}^{I_M} & prr_{S,L_2}^{I_M} & \cdots & prr_{S,L_N}^{I_M} \\ prr_{S,L_1}^{\varnothing} & prr_{S,L_2}^{\varnothing} & \cdots & prr_{S,L_N}^{\varnothing} \end{bmatrix} \tag{3}$$

Feature	Calculation	Category
(1) Average on-air time (T_{on})	$T_{on} = \frac{\sum_{k=1}^{min(N_S,N_E)}(e_k - s_k)\cdot T_s}{N_S}$	Temporal behaviour of interferers
(2) Occupancy level R_{occ}	$R_{occ} = \frac{\sum_{k=1}^{min(N_S,N_E)}(e_k - s_k)}{\sum_{i=1}^{minN_S,N_E}(s_{i+1} - s_i)}$	
(3) Energy span during interference (E_s)	$E_s = max(X^{(busy)}) - min(X^{(busy)})$	Energy characteristics
(4) Energy level during interference (E_l)	$E_l = \frac{\sum_{k=1}^{min(N_S,N_E)} X_k^{(busy)}}{N_S}$	
(5) Energy variance during interference (E_v)	$E_v = \frac{\sum_{k=1}^{min(N_S,N_E)}(X_k^{(busy)} - E_l)^2}{N_S}$	
(6) Peak to Average Power Ratio ($PAPR$)	$PAPR = max(X^{(busy)}) - E_l$	

TABLE I: Features Utilized for Interference Identification.

where $\{L_1, ..., L_N\}$ is $S's$ outbound link set, denoted as $\mathbf{L_S}$, and $\{I_1, ..., I_M\}$ are the interferences in the network environment. $prr_{S,L_n}^{I_m}$ means the PRR of Link L_n under interference I_m. In addition, the *LinkMap* also maintains a entry (the last entry) for the case that there is no interference. Using the *LinkMap*, the sender S can obtain PRRs of the outbound links under a specific interference. For example, if S detects I_m, it extracts the entry relevant to I_m (i.e. prr_{S,L_1}^{I}, ..., prr_{S,L_n}^{I}), and fed it to data transmission component (which will be discussed in Section VII) for further decision making.

Measurement in practice. Initially, the *LinkMap* is empty and thus when a new interference is detected by the interference identification component, the sender has to rely on burst probing, which is used in [11], to measure PRRs of the outbound links in $\mathbf{L_S}$. The reason to use burst probing, instead of classical periodic probing, is that we need to measure the up-to-date PRRs of links under the current interference, which cannot be captured by periodic probing that utilizes history information for PRR estimation. In the meantime, to avoid the collision between the interference and the ACK, the ACKs are piggybacked on the future normal data traffic from the receivers instead of replied immediately. After receiving the ACKs, the sender can compute the PRR of each link in $\mathbf{L_S}$ under the new interference, and add it to the *LinkMap*.

In addition, to keep the *LinkMap* up-to-date, the sender also updates *LinkMap* via normal data traffic. Specifically, if the sender selects Link L_n as the best link and transmits a series of packets through L_n under a certain interference I_m, it would obtain a PRR for this transmission task denoted as $prr_{S,L_n}^{I_{m\_new}}$. Then $prr_{S,L_n}^{I_m}$ can be updated using weighted moving average as follows:

$$prr_{S,L_n}^{I_m} = \theta \cdot prr_{S,L_n}^{I_m} + (1-\theta) \cdot prr_{S,L_n}^{I_{m\_new}}. \quad (4)$$

where θ is a tunable parameter which is set as 0.9 in our implementation.

B. Modeling the White Space Pattern

The Pareto Model. Although the sender has opportunities to transmit data when the channel is busy, the ACK must arrive at the sender when the channel is temporary idle (i.e., in the white space of the interference), otherwise collisions will occur. Thus the arrival process of the white space of the interference should be carefully modeled for controlling the ACK transmissions of ZigBee in presence of interference. As shown in [4, 25], the network traffic typically exhibits the self-similar [1] nature. Thus we use the Pareto model, the most widely adopted model to describe the self-similarity phenomenon, to fit the distribution of the white space as follows:

$$P(x > t) = \begin{cases} (\frac{\alpha_w}{t})^{\beta_W} & t > \alpha_W \\ 1 & otherwise \end{cases} \quad (5)$$

where $P(x > t)$ presents the probability that the length of the coming white space is larger than t. α_W and β_W are the scale and shape of the Pareto model respectively. Specifically, α_W can be given by the minimum of the white space length, and β_W is given by $\frac{\lambda_W}{\lambda_W - \alpha_W}$, where λ is the average length of the white space. In addition, we can also model the length of black space (i.e., the white space interval) and the white/black period using Pareto model with parameter (α_B, β_B) and (α_C, β_C).

Measurement of α and β. When a new interference is detected by the identification component, the sender samples the channel and measures the interval between two interference signals in order to build the white space model. Since the on-air time of ZigBee ACK is at least $200\mu s$ (accounting for the software overhead), thus only the interval longer than $200\mu s$ is considered as a sample of white space. The spaces between white spaces are treated as black spaces. Then α and β can be calculated based on the minimum and average value of the white/black segments. The size of the sampling window is set as $30ms$, which is long enough to ensure the accuracy of the Pareto model.

The value of α and β, together with the *LinkMap*, are stored in a **ILTable** (interference-link table) which provides fine-grained link and interference information (i.e. PRRs and (α, β)) to data transmission component (Section VII) for further decision making. We set a timer T_m $(1 \leq m \leq M)$ for each interferer I_m, which records the duration since the last appearance of I_m. T_m would be reset to zero when I_m is detected. If T_m exceeds a predefined lifetime (such as half an hour), I_m would be considered as moved away or turned OFF, and thus entry 'I_m' would be deleted from the **ILTable**.

VII. Interference-Adaptive Data Transmission

Target of the data transmission component is to adaptively select the link with the best quality under the detected inter-

[1]The self-similarity is a feature of arrival process with heavytailed or power law distributed inter-arrival time[4, 25].

ference and then schedule the transmission of data and ACKs for high channel utilization and low collision probability.

A. Concurrent Transmission v.s. Backoff Transmission

Before a transmission task, the sender should first determine the transmission mode, which refers to the option of concurrent transmission (denoted as H_c) or backoff transmission (denoted as H_c) through the selected best link. Particularly, concurrent transmission may improve the throughput, but suffers from low energy efficiency for retransmissions due to the relatively low PRR. On another hand, backoff transmission enables a sender to circumvent the interference, which brings better PRR but suffers from poor throughput. To this end, we propose an adaptive transmission method, which enables a node to predict the transmission capability and energy efficiency of each transmission mode, and then make a decision based on the predict result.

We define the transmission capability as the expected number of successfully transmitted packet in one white/black period. Thus under a specific interference I_m, a sender can estimate the transmission capability of concurrent transmission and backoff transmission as follows:

$$\begin{aligned} C_c &= \sum_{i=0}^{N_c}\sum_{k=0}^{i}\binom{\text{N-c-N-b}}{k}\cdot(prr_{S,L_c}^{I_m})^k\cdot(1-prr_{S,L_c}^{I_m})^{\text{N-c-N-b-k}} \\ &\quad+\sum_{i=0}^{N_c}\sum_{k=0}^{i}\binom{N_b}{i-k}\cdot(prr_{S,L_c}^{\varnothing})^{i-k}\cdot(1-prr_{S,L_c}^{\varnothing})^{\text{N-b-(i-k)}}. \\ C_b &= \sum_{i=0}^{N_b}\binom{N_b}{i}\cdot prr_{S,L_b}^{\varnothing}\cdot(1-prr_{S,L_b}^{\varnothing})^{N_b-i}. \end{aligned} \tag{6}$$

where C_c and C_b are the transmission capabilities for concurrent and backoff transmission, respectively. L_c is the best link for the sender S to transmit packets concurrently with I_m, which is selected based on the entry 'I_m' in the *LinkMap*. L_b is the best link for backoff transmission, which is selected based on the entry '$\varnothing$' in the *LinkMap*. N_c and N_b are the maximum number of successfully transmitted packets for concurrent and backoff transmission in one white/black period. Assuming that the time for transmitting one packet is T_p, then N_c and N_b can be estimated as $N_c = \frac{T_{black}+T_{white}}{T_p}$ and $N_c = \frac{T_{white}}{T_p}$. T_{black} and T_{white} can be obtained from the inverse Pareto function as $T_{black} = P_b^{-1}(p) = \frac{\alpha_B}{\sqrt[\beta_B]{p}}$ and $T_{white} = P_w^{-1}(p) = \frac{\alpha_W}{\sqrt[\beta_W]{p}}$, which give the length of black space and white space for a given confidence interval p.

In addition, assume the energy cost for transmitting one packet is E_{trans}. Thus the energy efficiency of concurrent transmission and backoff transmission can be calculated as $E_c = \frac{C_c}{E_{trans}\cdot N_c}$ and $E_b = \frac{C_b}{E_{trans}\cdot N_b}$, respectively.

Then, given a predefined energy efficient constraint (denote as E_{th}), the sender would make a choice between concurrent transmission and backoff transmission. Specifically, if both the two transmission modes satisfy $E > E_{th}$, we choose

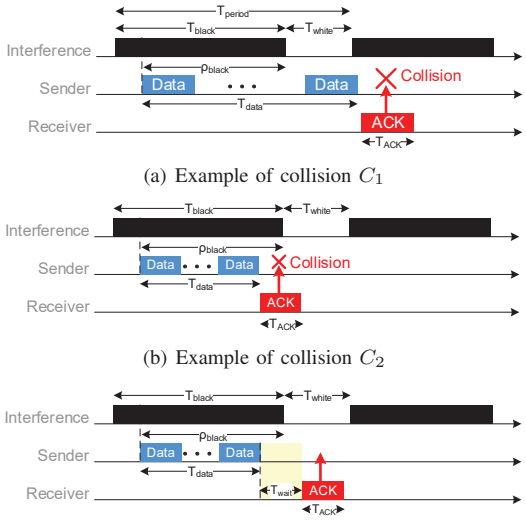


Fig. 6: Two Collision Cases and the Solution.

concurrent transmission or backoff transmission by comparing their transmission capacity as follows:

$$\begin{cases} H_c & C_c > C_b \\ H_b & C_c \le C_b \end{cases} \tag{7}$$

otherwise, we choose H_c or H_b by comparing their energy efficiency:

$$\begin{cases} H_c & E_c > E_b \\ H_b & E_c \le E_b \end{cases} \tag{8}$$

If mode H_c is selected, the sender would immediately transmit packets through link L_c. Otherwise, the sender would pause the transmission and makes a random backoff, and then transmit packets through link L_b.

B. Schedule of Data and ACKs

Target of the schedule mechanism is to carefully control the transmission of data packets and ACKs to achieve high channel utilization and low collision probability. Collisions between the ACK and the interference can be classified into two cases:

i) the collision with the black space in the next black/white period, which caused by a long data period T_{data}, as shown in Figure 6(a). We denote such collision as C_1.

ii) the collision with the black space in the current black/white period, which caused by a short data period T_{data}, as shown in Figure 6(b). We denote such collision as C_2.

Figure 6(c) illustrates an example of the scheduling mechanism. To avoid the collision C_1, the sender should first set a constraint on the length of the data period. In addition, to avoid the collision C_2 the sender has to predict the arrive time of the next white space, and then notifies the receiver to wait T_{wait} after receiving the last data packet and then reply the ACK. This back-off saves the ACK from colliding with the current interference.

1) Limiting the length of data period: As shown in Figure 6(a), to avoid collision C_1, the length of data period T_{data} must satisfies that:

$$T_{data} < \rho_{black} + T_{white} \tag{9}$$

where ρ_{black} is the remaining duration of T_{black} upon the start of data transmission. It is easy to understand that C_1 occurs if the first data packet arrives $T_{period} - T_{data}$ later than the start of the current black space (T_{period} is the length of one white/black period as shown in Figure 6(a)). Assume that ρ_{black} is uniformly distributed over the entire black space, the probability of C_1 can be estimated as $\min\{\frac{T_{data}}{T_{period}}, 1\}$.

Based on the proposed pareto model in Section VI, the expected probability of C_1 is given by

$$P_{C_1}(T_{data}) = 1 - \frac{1}{\beta_C}(\frac{\alpha_C}{T_{data}})^{\beta_C - 1} \tag{10}$$

Given a specific probability threshold C_{th}, T_{data} must satisfies:

$$P_{C_1}(T_{data}) < C_{th} \tag{11}$$

By solving Equation (11), we set a constrain on T_{data} as follows:

$$T_{data} < T_{data}^{(max)} = \frac{\alpha}{(\beta \cdot (1 - C_{th}))^{\frac{1}{\beta - 1}}} \tag{12}$$

Therefore, before a sender starts a sending task with n data packet, it should first compare the required T_{data} with $T_{data}^{(max)}$. If $T_{data} > T_{data}^{(max)}$, the sender has to reduce the packet number in the current white/black period utile the T_{data} satisfy the constraint shown in Equation 12.

2) Adjusting the arrive time of ACKs: As shown in Figure 6(b), C_2 occurs if $T_{data} < \rho_{black}$. Thus in this scenario, the sender has to estimate the arrive time of the next white space based on the Pareto model in Section VI, and notifies the receiver to wait T_{wait} before replying the ACK. Specifically, T_{wait} should satisfies:

$$T_{data} + T_{wait} > \rho_{black} \tag{13}$$

Thus, the probability of C_2 can be estimated as $\min\{\frac{T_{black} - (T_{data} + T_{wait})}{T_{black}}, 1\}$. Then the expected collision probability of C_2 is given by

$$P_{C_2}(T_{wait}) = \frac{1}{\beta_B}(\frac{\alpha_B}{T_{data} + T_{wait}})^{\beta_B - 1} \tag{14}$$

Given a collision probability threshold C_{th}, T_{waite} must satisfies:

$$P_{C_2}(T_{wait}) < C_{th} \tag{15}$$

By solving Equation (15), we set a constrain on T_{wait} as follows:

$$T_{wait} > T_{wait}^{(min)} = \frac{\alpha_B}{(\beta_B \cdot C_{th})^{\frac{1}{\beta_B - 1}}} - T_{data} \tag{16}$$

$T_{wait}^{(min)}$ is piggybacked on the last data packet sent to the receiver. If $T_{wait}^{(min)} \leq 0$, the receiver can rely the ACKs immediately. Otherwise, it has to wait at least $T_{wait}^{(min)}$ before replying the ACKs. To note that, Smoggy-Link uses one ACK to reply all the data packets transmitted in the current white/black period. To do that, each data packet has to piggyback its sequence number and the number of the packets to be transmitted in this period.

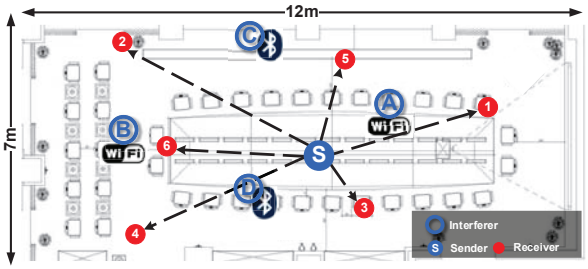


Fig. 7: Layout of the Experiment Setup.

VIII. Experimental Evaluation

In this section, we present the evaluation results of Smoggy-Link. We implemented Smoggy-Link in TinyOS 2.x on TelosB motes equipped with 802.15.4 radios, and evaluate its performance in terms of interference identification accuracy, link estimation accuracy, throughput and retransmission cost.

A. Experimental Setup

The experiment takes place in a large conference room, and the deployment scenario is depicted in Figure 7. We randomly deploy 7 nodes in the room, where one is the sender and the rests are receivers. Transmission power of the sender is set to level 5. The experiment lasts for 60 min, during which the sender periodically broadcasts 200-byte payload packets at a rate of 10 packets per second, and the receivers listen and record the sequence numbers of the received packets. All experiments are conducted in the 12th channel.

The locations of the interferers are highlighted in Figure 7 as circles denoted as A, B, C, and D. Specifically, two Wi-Fi interferers are located at location A (denote as $Wi - Fi^{(1)}$) and B (denote as $Wi - Fi^{(2)}$), respectively. Two Bluetooth interferers are located at location C (denote as $Bluetooth^{(1)}$) and D (denote as $Bluetooth^{(2)}$), respectively. During the experiment, we periodically turn ON each interferer to generate interference signal for $10min$. The ON/OFF pattern of the interferers is shown in Figure 8 (a).

Compared schemes:

- *Smoggy-Link:* is the proposed design in this paper.
- *Beacon+CSMA off:* transmission method with CSMA disabled using beacon-based link estimation.
- *Beacon+CSMA on:* transmission method with CSMA enabled using beacon-based link estimation.

Specifically, Smoggy-Link and Beacon+'CSMA off' adopt interference concurrency, while Beacon+ 'CSMA on' adopts interference avoidance.

Identified as	A	B	C	D
A. $Wi-Fi^{(1)}$	**94.3**	3.4	0.2	2.1
B. $Wi-Fi^{(2)}$	3.2	**95**	0.7	1.1
C. $Bluetooth^{(1)}$	4.3	3.4	**90.1**	2.2
D. $Bluetooth^{(2)}$	3.5	2.7	5.3	**90.9**

TABLE II: Confusion Matrix of the Interference Identification Component.

B. Interference Identification Accuracy

We first evaluate the performance of Smoggy-Link's interference identification component, which is particulary important for the performance of the whole system. During the experiment, the sender collects the RSSI segments of the interferences and identify it. The interference identification is triggered every 6s (i.e., before every transmission task).

Table II shows the confusion matrix for the interference identification component. Each element in the matrix corresponds to the fraction of the interference in the row that is identified as the interference in the column. The table tells that i) Smoggy-Link achieves a max identification accuracy of 95% and an average accuracy of 92.35%, with a standard deviation of 2.26%; and ii) the average accuracy for identifying bluetooth (about 90%) is lower than that of WIFI (about 95%). This is because the energy level of Bluetooth is not so stable as Wi-Fi. The above results show that the interference identification method can distinguish different interferences with high accuracy by using only RSSI segments.

C. Link Estimation Accuracy

We evaluate link estimation performance of Smoggy-Link by observing the link selection behavior of the sender. As shown in the previous sections, link selection is highly depended on the current interference in the network. The sender would select the best link based on the interference identification result. Figure 8(a) shows the ground trues of the ON/OFF pattern of the interferers, and Figure 8(b) shows a comparison between the corresponding link selection behaviors of Smoggy-Link and beacon based link estimator.

In Figure 8(b), the matrix-color mapping table shows the PRR of different links (a lighter color means a higher PRR). The square and circle markers show the link selection traces of Smoggy-Link and beacon-based estimator, respectively. As shown in the figure, under the interference free environment (period 1-10 min), both Smoggy-Link and beacon-based estimator select Link 3 for data transmission. However, once the $Wi-Fi^{(1)}$ is turned ON, the quality of Link 3 decreases sharply. Smoggy-Link can quickly adapt to this change by switching to Link 2 which exhibits higher PRR, and thus successfully circumvents the interference it would else have experienced. However, beacon based link estimator always select Link 4 (with a probability of 39%) instead. We can observe similar behaviors when the $Wi-Fi^{(2)}$, $Bluetooth^{(1)}$ and $Bluetooth^{(2)}$ is ON. The reason that Smoggy-Link's performance is better than beacon based estimator is that Smoggy-Link makes link selection based on the up-to-date

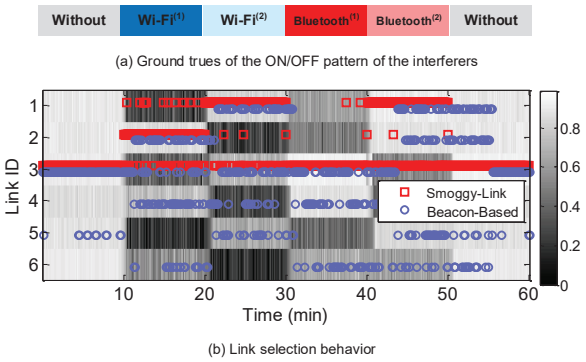


Fig. 8: Link Selection Behavior Comparison.

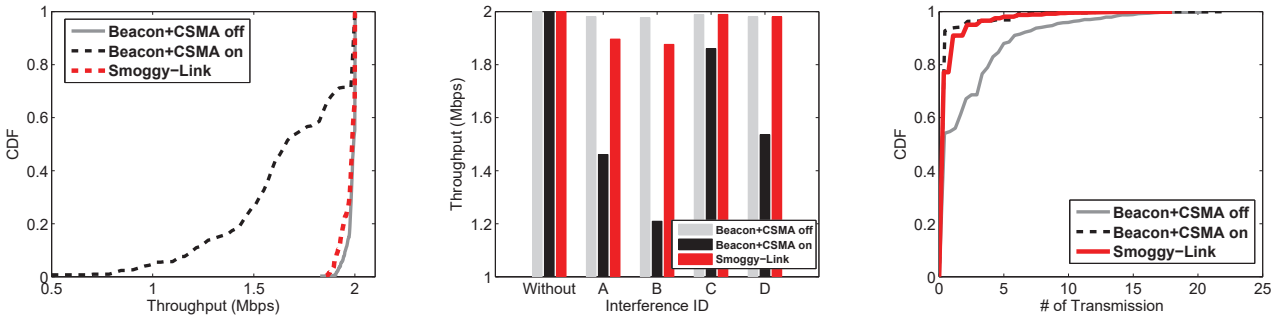
interference information, while beacon-based estimator relies on the outdated packet reception history. In addition, collision between ACKs and the interference would undermine the performance of the beacon based estimator.

D. Network Performance

Throughput. Figures 9(a) and 9(b) show the comparison of different schemes on throughput. We quantify the throughput by counting the total packets received by the 6 receivers during a period of 10 seconds windows. As shown in Figure 9(a), the throughput of the interference avoidance method (i.e., Beacon+'CSMA on') is far less than that of interference concurrency methods (i.e., Beacon+'CSMA off' and Smoggy-Link) because the former would suppress the opportunities of concurrent transmissions. Beacon+'CSMA off' attains the largest throughput among the three schemes. This is because Beacon+'CSMA off' can maximally exploit the concurrent opportunities by completely turn off the CSMA, which however leads to poor PRR and thus high energy consumption for packet retransmission. Compared with Beacon+'CSMA on', Smoggy-Link can improve the throughput by 33%, because Smoggy-Link can quickly confirm the feasibility of exploiting concurrent opportunity by considering the PRR under the current interference.

In addition, Figure 9(b) shows how different interferences impact the throughput of ZigBee. We can see that, throughput of Smoggy-Link only decreases slightly even under strong interferences (such as Interference A and B). While the Beacon+'CSMA on' suffers serious decline on throughput, especially under strong interference (with a degradation of 45% compared with the scenario that there is no interference).

Retransmission. Although concurrent transmission is adopted in Smoggy-Link, indicating the PRR will decrease in some degree, the retransmission count is comparable with that of interference avoidance approaches, shown in Figure 9(c). Figure 9(c) shows that as Smoggy-Link can adaptively select the strongest link under different interference, its cost for retransmission is obviously less than that of Beacon+'CSMA off' and almost equal to that of Beacon+'CSMA on'.



(a) Distribution of throughput (b) Throughput under different interference (c) Distribution of retransmission

Fig. 9: Network Performances Comparison.

IX. Conclusion

In this paper, we propose Smoggy-Link to exploit potential concurrency for adaptive transmission under interference. Smoggy-Link maintains a link model which can estimate the outbound link's conditional packet reception ratio in realtime under different interference and select the strongest link for concurrent transmission. In addition, Smoggy-Link can intelligently schedule the transmission of data and ACKs to achieve both high channel utilization and low ACK collision probability. We implement and evaluate a prototype of our approach with TinyOS and TelosB motes. The evaluation results show that Smoggy-Link has consistent improvements in both throughput and packet reception ratio under interference from various interferers.

Acknowledgment

This work was supported by National Basic Research Program (973 program) under Grant of 2014CB347800, National Science Fund for Excellent Young Scientist No.61422207, The NSFC (61672428, 61272461, 61672320, 61602381, 61572402).

References

[1] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the RF Smog: Making 802.11n Robust to Cross-technology Interference. In *ACM SIGCOMM*, pages 170–181, 2011.

[2] X. Zhang and K. G. Shin. Enabling Coexistence of Heterogeneous Wireless Systems: Case for Zigbee and Wifi. In *ACM MobiHoc*, pages 3967–3974, 2011.

[3] B. Radunovic, R. Chandra, and D. Gunawardena. Weeble: Enabling Low-power Nodes to Coexist with High-power Nodes in White Space Networks. In *ACM CONEXT*, pages 205–216, 2012.

[4] J. Huang, G. Xing, G. Zhou, and R. Zhou. Beyond Co-existence: Exploiting WiFi White Space for ZigBee Performance Assurance. In *IEEE ICNP*, pages 305–314, 2010.

[5] Y. Yan, P. Yang, X. Li, Y. Tao, L. Zhang, and L. You. ZIMO: Building Cross-Technology MIMO to Harmonize ZigBee Smog with WiFi Flash without Intervention. In *ACM MobiCom*, pages 465–476, 2013.

[6] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat. Learning to Share: Narrowband-friendly Wideband Networks. In *ACM SIGCOMM*, pages 147–158, 2008.

[7] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng. Supporting Demanding Wireless Applications with Frequency-agile Radios. In *USENIX NSDI*, pages 65–80, 2010.

[8] R. Musaloiu-E and A. Terzis. Minimising the Effect of WiFi Interference in 802.15.4 Wireless Sensor Networks. *International Journal of Sensor Networks*, 3(1):43–54, 2008.

[9] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, 18(3):535 – 547, 2000.

[10] X. Zhang and K. G. Shin. Gap Sense: Lightweight Coordination of Heterogeneous Wireless Devices. In *IEEE INFOCOM*, pages 3094 – 3101, 2013.

[11] S. M. Kim, S. Wang, and T. He. cETX: Incorporating Spatiotemporal Correlation for Better Wireless Networking. In *ACM SenSys*, pages 323–336, 2015.

[12] J. Polastre, J. Hill, and D. Culler. Versatile Low Power Media Access for Wireless Sensor Networks. In *ACM SenSys*, pages 95–107, 2004.

[13] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC:Ashort Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In *ACM SenSys*, pages 307–320, 2006.

[14] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy. CrossZig: Combating Cross-Technology Interference in Low-power Wireless Networks. In *ACM/IEEE IPSN*, pages 1–12, 2016.

[15] Z. Li, Y. Xie, M. Li, and K. Jamieson. Recitation: Rehearsing Wireless Packet Reception in Software. In *ACM MobiCom*, pages 291–303, 2015.

[16] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu. Path Reconstruction in Dynamic Wireless Sensor Networks Using Compressive Sensing. In *ACM MobiHoc*, pages 297–306, 2014.

[17] X. Zheng, J. Wang, W. Dong, Y. He, and Y. Liu. Bulk Data Dissemination in Wireless Sensor Networks: Analysis, Implications and Improvement. *IEEE Transactions on Computers*, 65(5):1428–1439, 2016.

[18] J. Lu and K. Whitehouse. Flash Flooding: Exploiting the Capture Effect for Rapid Flooding in Wireless Sensor Networks. In *IEEE INFOCOM*, pages 2491–2499, 2009.

[19] X. Zhang and K. G. Shin. Chorus: Collision resolution for efficient wireless broadcast. In *IEEE INFOCOM*, pages 1747–1755, 2011.

[20] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *ACM/IEEE IPSN*, pages 73–84, 2011.

[21] X. Ji, Y. He, J. Wang, K. Wu, K. Yi, and Y. Liu. Voice over the Dins: Improving Wireless Channel Utilization with Collision Tolerance. In *IEEE ICNP*, pages 1–10, 2013.

[22] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu. ZiSense: Towards Interference Resilient Duty Cycling in Wireless Sensor Networks. In *ACM SenSys*, pages 119–133, 2014.

[23] F. Hermans, O. Rensfelt, T. Voigt, and P. Gunningberg. SoNIC: Classifying Interference in 802.15.4 Sensor Networks. In *ACM/IEEE IPSN*, pages 55–66, 2013.

[24] A. Hithnawi, H. Shafagh, and S. Duquennoy. TIIM: Technology-independent Interference Mitigation for Low-power Wireless Networks. In *ACM/IEEE IPSN*, pages 1–12, 2015.

[25] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-similar Nature of Ethernet Traffic. In *ACM SIGCOMM*, pages 183–193, 1993.