BiCord: Bidirectional Coordination among Coexisting Wireless Devices

Zihao Yu
Tsinghua University
Beijing, China
zh-yu17@mails.tsinghua.edu.cn

Pengyu Li
Tsinghua University
Beijing, China
lpy18@mails.tsinghua.edu.cn

Carlo Alberto Boano
Graz University of Technology
Graz, Austria
cboano@tugraz.at

Yuan He
Tsinghua University
Beijing, China
heyuan@tsinghua.edu.cn

Meng Jin
Tsinghua University
Beijing, China
mengj@mail.tsinghua.edu.cn

Xiuzhen Guo
Tsinghua University
Beijing, China
guoxz16@mails.tsinghua.edu.cn

Xiaolong Zheng
Beijing University of Posts and Telecommunications
Beijing, China
zhengxiaolong.zxl@gmail.com

***Abstract*—Cross-technology interference is a major threat to the dependability of low-power wireless communications. Due to power and bandwidth asymmetries, technologies such as Wi-Fi tend to dominate the RF channel and unintentionally destroy low-power wireless communications from resource-constrained technologies such as ZigBee, leading to severe coexistence issues. To address these issues, existing schemes make ZigBee nodes individually assess the RF channel's availability or let Wi-Fi appliances blindly reserve the medium for the transmissions of low-power devices. Without a two-way interaction between devices making use of different wireless technologies, these approaches have limited scenarios or achieve inefficient network performance. This paper presents BiCord, a bidirectional coordination scheme in which resource-constrained wireless devices such as ZigBee nodes and powerful Wi-Fi appliances coordinate their activities to increase coexistence and enhance network performance. Specifically, in BiCord, ZigBee nodes directly request channel resources from Wi-Fi devices, who then reserve the channel for ZigBee transmissions on-demand. This interaction continues until the transmission requirement of ZigBee nodes is both fulfilled and understood by Wi-Fi devices. This way, BiCord avoids unnecessary channel allocations, maximizes the availability of the spectrum, and minimizes transmission delays. We evaluate BiCord on off-the-shelf Wi-Fi and ZigBee devices, demonstrating its effectiveness experimentally. Among others, our results show that BiCord increases channel utilization by up to 50.6% and reduces the average transmission delay of ZigBee nodes by 84.2% compared to state-of-the-art approaches.**

***Index Terms*—Coexistence, Cross-technology interference, Cross-technology communication, Device coordination**

I. INTRODUCTION

With the rapid growth of the Internet of Things (IoT), an increasing number of wireless devices is crowding the unlicensed ISM bands, causing severe cross-technology interference (CTI). This problem is especially serious in the 2.4 GHz band, where several pervasive technologies such as Wi-Fi, Bluetooth, and ZigBee share the same frequencies [1].

To coordinate access to the medium across devices using different technologies, wireless systems typically adopt CSMA/CA: when detecting ongoing activities in the RF channel by means of energy sensing, a device defers its own transmissions to avoid a collision. In real-world scenarios, however, due to the power and bandwidth asymmetries of different wireless technologies, devices may not correctly hear each other's transmissions, which may result in an unfair and inconsistent channel allocation. For example, Wi-Fi devices are usually unable to sense surrounding ZigBee traffic because of the large difference in transmission power and the receiver sensitivity[1]: as a result, ZigBee nodes are highly vulnerable to the CTI caused by nearby Wi-Fi devices, and may suffer a decrease in packet reception rate by up to 85% [2].

This state of affairs calls for techniques that enable efficient coordination among devices making use of different wireless technologies. Even if a low-power wireless device (e.g., a ZigBee node) faces an intrinsic disadvantage in coexisting and contending the channel access with a more powerful appliance (e.g., a Wi-Fi device), it should still be possible for both parties to interact and cooperatively share the medium.

Gauging channel availability is not enough. Traditional approaches let low-power wireless devices locally measure, assess, and predict the availability of an idle RF channel (e.g., the occurrence of white spaces free of Wi-Fi traffic [3], [4]), so as to occupy the medium accordingly. Although this mechanism allows to passively avoid Wi-Fi interference, it performs poorly with high traffic dynamics, as ZigBee nodes are unable to obtain *accurate* information about the traffic pattern of nearby Wi-Fi devices using local observations only.

Unidirectional information transfer is insufficient. Recent approaches allow communication from powerful appliances to low-power devices. For example, using ECC [6], Wi-Fi devices can broadcast a cross-technology message to notify nearby ZigBee nodes about an upcoming white space that may be exploited for their transmissions. However, without knowing

[1]The transmission power of Wi-Fi and ZigBee devices is typically in the order of 20 and 0 dBm, respectively. To detect ongoing transmissions based on the received signal strength indicator (RSSI), ZigBee nodes typically consider a channel busy when RSSI$\geq$-82 dBm, while Wi-Fi devices commonly use a threshold of -70 dBm.

the requirements of the surrounding ZigBee nodes, a Wi-Fi device can only *blindly* allocate the channel for ZigBee nodes, which ultimately leads to an ineffective channel allocation. Indeed, ZigBee nodes can only wait for notifications and cannot explicitly ask for channel access when actually needed.

Need for bidirectional coordination. To make an efficient use of the shared RF spectrum, we need to enable communication from the low-power wireless devices to the more powerful appliances as well, such that the former can share their channel access requirements and such that the latter can reserve the channel on-demand. In principle, ZigBee nodes could send a message to nearby Wi-Fi devices using existing cross-technology communication (CTC) schemes to inform them that they need to access the channel. However, CTC schemes from a constrained to a more powerful wireless technology face several limitations due to power and bandwidth asymmetries[2]. For example, physical-layer CTC schemes such as LEGO-Fi [7] and SymBee [8] enable a data exchange from ZigBee to Wi-Fi, but require modifications to the radio and hence do not run on off-the-shelf devices. Similarly, other solutions based on packet-level modulation [9], [10] incur large delays that are unsuitable for the design of low-power wireless medium access control schemes, as detailed in Sec. III. For these reasons, state-of-the-art CTC solutions cannot be directly reused to enable a low-latency bidirectional communication between off-the-shelf Wi-Fi and ZigBee devices: this prevents us from building an effective channel allocation scheme.

Our contributions. This work introduces BiCord, a bidirectional coordination scheme for devices using heterogeneous wireless technologies that allows an efficient allocation of the RF channel. In BiCord, constrained wireless devices can request more powerful appliances in their surroundings to allocate the channel by means of cross-technology signaling, i.e., by transmitting control information that can be decoded also by appliances using an incompatible physical layer. The more powerful devices receive the control information, learn at runtime the transmission patterns of nearby low-power nodes, and issue, in response to a cross-technology signal, a white space of sufficient length. This way, the more powerful devices can allocate the channel on-demand based on the requirements of surrounding low-power devices and avoid to over- or under-provision channel resources due to uninformed decisions.

After reviewing related work (Sec. II) and describing the motivation for bidirectional coordination (Sec. III), we present BiCord's design principles (Sec. IV) and showcase its operations with an implementation on commercial Wi-Fi and ZigBee devices. This paper makes the following contributions:

- We develop a method allowing ZigBee nodes to directly inform nearby Wi-Fi devices about their need to access the channel. The method is reliable, efficient, and has little impact on the performance of Wi-Fi devices (Sec. V).
- We let the more powerful devices learn the transmission patterns of surrounding low-power nodes and adaptively generate white spaces in response to their needs. This provides low-latency and on-demand service to low-power devices and increases channel utilization (Sec. VI).
- We run BiCord on off-the-shelf ZigBee and Wi-Fi devices, discuss selected implementation details (Sec. VII), and evaluate its effectiveness experimentally (Sec. VIII).

We finally conclude and summarize our work in Sec. IX.

[2]Wi-Fi devices operating in the 2.4 GHz band typically employ a channel bandwidth of 20 or 40 MHz, whereas ZigBee channels have a bandwidth of 2 MHz only as specified by IEEE 802.15.4.

II. Related Work

Cross-technology coexistence. In CTI scenarios, it is difficult for low-power devices to contend channel access against more powerful devices or to handle interference using classical approaches [11]–[13]. For example, Wi-Fi's transmission power is roughly 1000 times higher than that of ZigBee, resulting in Wi-Fi devices often overlooking and corrupting ZigBee traffic. Moreover, due to the limited capability of their radios, ZigBee nodes spend more time than Wi-Fi devices in sensing the channel prior transmission, and are often preempted by faster Wi-Fi devices when switching from carrier sensing to transmission mode [14], which results in high packet loss rates.

Early works addressing the CTI problem aim to *recover* from interference [15]–[18]. Although effective, these methods usually require additional spectrum resources (e.g., extra coding bits in the packets). Another class of solutions aims to *avoid* interference [4], [19]–[21]. In such schemes, the challenge is to provide ZigBee nodes with fine-grained information about the interference patterns, as the latter can be used as an indicator of the transmission schedule [4], [20] or the coding strategy [22]. Before the development of CTC, the only way to get such information was to analyze the channel hints that were left by the interfering signals [4], [20], [21]. However, due to the intrinsic variability of the interference patterns, it is difficult for ZigBee nodes to accurately predict the channel occupancy and guarantee collision-free transmissions, which inevitably leads to poor channel utilization. Note that in both recovery- and avoidance-based methods, ZigBee nodes do not interact or coordinate with nearby Wi-Fi devices.

Recently, with the development of CTC schemes, ZigBee nodes are able to directly communicate with surrounding devices (i.e., with the potential sources of interference). This allows to build more accurate channel coordination schemes and increase channel utilization. For example, in ECC [6], Wi-Fi devices generate white spaces and notify nearby ZigBee nodes about their existence through CTC. However, the performance of ZigBee depends on the arrangements made by Wi-Fi, i.e., on how often Wi-Fi devices reserve the white space and for how long. Given that Wi-Fi devices allocate the channel "blindly", they may not meet the requirements of ZigBee nodes, ultimately resulting in long transmission delays and ineffective channel utilization.

In this work, we address the main limitation of ECC (i.e., the unidirectionality of the data exchange) by proposing a bidirectional coordination scheme that allows to increase the coexistence among ZigBee and Wi-Fi devices.

Cross-technology communication. Several works have enabled CTC between Wi-Fi, BLE, and ZigBee devices (i.e., the most ubiquitous technologies using the 2.4 GHz band). Early works introduce CTC by means of packet-level modulation [9], [10], [23]–[28], which manipulates the packet as information carrier in order to build an accessible side channel. Due to the coarse-grained information, the throughput of packet-level CTC is limited to a few bps [23] or a few kbps [28]. In order to improve the data rate, recent works propose CTC by means of physical-layer emulation [5], [29]–[34], which aims to create compliance across diverse technologies and to build the CTC channel right at the physical layer. For example, WEBee [5] uses the Wi-Fi radio to emulate the standard time-domain signals of ZigBee and achieve a high throughput.

Despite these works, there is relatively little progress in CTC from a constrained to a powerful technology (e.g., from ZigBee to Wi-Fi). Enabling communication in this direction is extremely difficult due to power and bandwidth asymmetries, as well as to the low sensitivity of Wi-Fi receivers. First, the transmission power of ZigBee and the receiver sensitivity of Wi-Fi are rather low, which makes it difficult for Wi-Fi devices to reliably detect ZigBee signals [24], [27]. Furthermore, the channel bandwidth of Wi-Fi is much higher than that of ZigBee, which makes physical-layer emulation infeasible on commercial devices [7], [8]. Hence, existing CTC schemes from ZigBee to Wi-Fi are limited to packet-level modulation [9], which incurs large delays and is unsuitable for the design of channel coordination schemes.

In this work, we tackle these limitations and propose a cross-technology signaling scheme that initiates efficient channel coordination between ZigBee and Wi-Fi devices.

III. Motivation

A. The need for bidirectional coordination

To tackle the CTI problem, early works have proposed collision avoidance approaches to make ZigBee predict and stagger Wi-Fi traffic based on local channel assessment. These passive approaches do not allow to explicitly allocate the channel, such that a device has sufficient time to complete its transmissions, and are inherently prone to uncertainties from high traffic dynamics. Under random interference, the difficulty of modeling the interference leads to degraded performance of ZigBee (packet reception rate of 51% [6]), which inevitably leads to a poor channel utilization [15]–[17].

Recently, with the development of CTC schemes, more accurate channel coordination mechanisms are proposed to increase channel utilization. For example, in ECC [6], Wi-Fi devices voluntarily generate white spaces of a given length using CTS messages and explicitly notify nearby ZigBee nodes about their existence through CTC, as shown in Fig. 1. ZigBee nodes hence wait for Wi-Fi's cross-technology notification and start their transmissions in the provided white space: this way, the traffic of ZigBee and Wi-Fi devices is staggered.

However, as Wi-Fi devices do not know when ZigBee nodes have data to transmit or how much time they need to complete their transmissions, ECC periodically generates white spaces

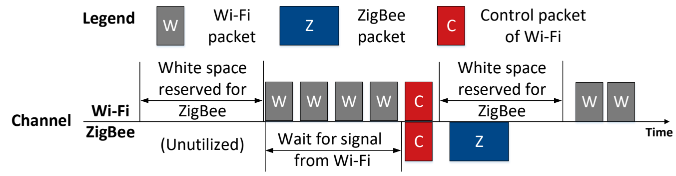


Fig. 1. In ECC, Wi-Fi devices signal the presence of a white space to nearby ZigBee nodes using CTC. However, the white spaces are of predefined length, as Wi-Fi devices do not know the requirements of surrounding ZigBee nodes on data transmission, which prevents an effective use of the shared channel.

of predefined length, which often leads to extended delays and/or underutilized channels. On the one hand, ECC may waste precious channel resources: Wi-Fi devices may reserve a white space when surrounding ZigBee nodes have no data to transmit (i.e., voluntarily generating a white space that is actually not useful). Similarly, Wi-Fi devices may over-provision the length of the white space, allocating the channel for ZigBee transmissions longer than necessary. On the other hand, ECC does not help decreasing the delay of ZigBee transmissions. Since bursts of ZigBee traffic may come at random times, the interval between two bursts may be shorter or larger than the one between two white spaces left by Wi-Fi. Hence, it may happen that a burst of ZigBee traffic has to wait for a significant amount of time until the Wi-Fi device issues a white space. Moreover, Wi-Fi devices may reserve a white space that is insufficient for ZigBee nodes to complete their data transfer. For example, one white space lasting $20ms$ can only accommodate the transmission of 3 consecutive ZigBee packets of 50bytes with acknowledgment (ACK). When a ZigBee node has more data to send in a burst (i.e., more packets, or packets with a larger payload), ZigBee has to scatter its transmissions into multiple white spaces, which leads to extended delays. To alleviate these problems, ECC proposes that Wi-Fi devices estimate the interval between ZigBee transmissions, and adjust the white space accordingly. However, this scheme relies on the assumption that ZigBee transmissions are exactly periodic and with a fixed length, which hardly holds true in the real-world [35]. Even worse, because of the missing feedback from the ZigBee nodes, the Wi-Fi devices may not generate a white space when it is actually needed, which could cause unbounded delays that are unacceptable for safety-critical ZigBee applications.

B. The challenges of bidirectional coordination

In this work, we explore how to enable a bidirectional coordination between Wi-Fi and ZigBee devices. As Wi-Fi devices use a higher power and tend to dominate the channel, it is easy to let them allocate channel resources and inform ZigBee nodes using physical-layer CTC [5]. However, it is difficult for ZigBee nodes to interact with Wi-Fi devices and let them know about their needs. ZigBee transmissions are usually overlooked by Wi-Fi devices, and it is infeasible for ZigBee nodes to emulate Wi-Fi signals using physical-layer CTC due to power and bandwidth asymmetries. Moreover, existing packet-level CTC schemes from ZigBee to Wi-Fi are unsuitable to build a channel coordination scheme. CTC

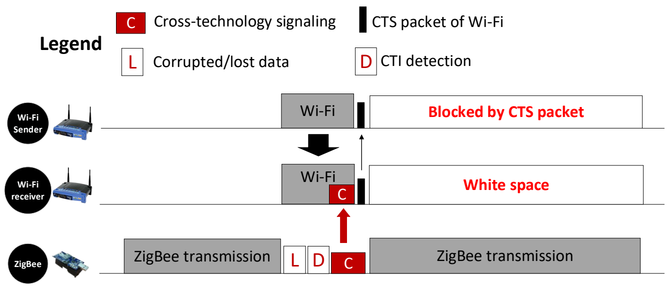


Fig. 2. In BiCord, ZigBee nodes sensing the presence of Wi-Fi traffic use cross-technology signaling to notify their need to access the channel. Nearby Wi-Fi devices understand the channel request and generate a white space.

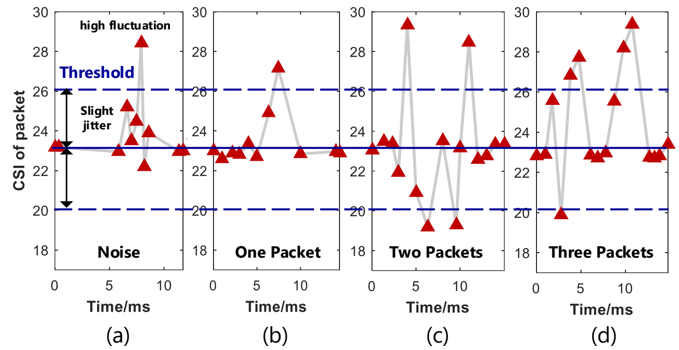


Fig. 3. CSI pattern in the presence of noise (a) and ZigBee signal for a number of control packets (b–d).

schemes such as FreeBee [23], are only effective in the presence of a clear channel, which is inefficient and even useless when Wi-Fi occupies the channel. Although recent CTC schemes like ZigFi [9] are able to perform CTC on a busy channel (e.g., when nearby Wi-Fi devices are operating), a tight synchronization between ZigBee nodes and Wi-Fi devices is needed. As a synchronization process would incur long delays, the benefits of the coordination scheme would be neutralized. Specifically, in recent CTC schemes from ZigBee to Wi-Fi, time is segmented into continuous time windows and the packet-level information is encoded or decoded in every time window. To transfer CTC data, it is necessary for Wi-Fi to synchronize with ZigBee and to find the start of each time window. In AdaComm [10], ZigBee nodes make use of a Barker code to synchronize with Wi-Fi devices, but the whole process is very slow ($\approx$ 110 ms). Such a delay is huge in comparison to the typical white space length requested by ZigBee nodes: five packets of 50 bytes each including ACK are transmitted in about 30 ms [6].

IV. BiCord: Design overview

To tackle these challenges, we propose BiCord, a bidirectional coordination scheme that enables a two-way interaction between Wi-Fi and ZigBee devices. To initiate the coordination, BiCord uses a cross-technology signaling method enabling ZigBee nodes to directly interact with Wi-Fi. Such a method allows an efficient and reliable CTC from ZigBee to Wi-Fi *without* the need of synchronization, which was a fundamental challenge highlighted in Sec. III-B.

Fig. 2 sketches the basic operations of BiCord. After a ZigBee node suffers from data corruption or a low packet reception rate under Wi-Fi interference, it first performs CTI detection (detailed in Sec. VII-A), identify the Wi-Fi traffic, and then conduct cross-technology signaling to inform nearby Wi-Fi devices about its need to access the channel. Specifically, ZigBee nodes send control packets overlapping with Wi-Fi transmissions, such that a Wi-Fi receiver detects the presence of ZigBee traffic by means of CSI (channel state information) analysis. Upon detection of ZigBee transmissions, a Wi-Fi device can broadcast a clear-to-send (CTS) packet so that surrounding Wi-Fi devices stop transmitting for a given period of time, hence generating a white space protecting ZigBee packets.

BiCord is mainly composed of two modules: *cross-technology signaling* and *adaptive white space allocation*. The cross-technology signaling module enables ZigBee nodes to transmit control information to Wi-Fi appliances: this way, a Wi-Fi device can quickly receive a one-bit control information indicating the existence of ZigBee traffic. Note that the signaling method happens in the side channel, such that the ongoing Wi-Fi traffic is minimally impacted, as detailed in Sec. V.

In response to ZigBee's control information, a Wi-Fi device provides white space for ZigBee transmissions. To increase channel utilization, the Wi-Fi device infers the length of the white space required by the ZigBee node and then adapts the white space allocation, as detailed in Sec. VI.

Note that the Wi-Fi device is not forced to allocate a white space, but can decide to defer its transmissions (and broadcast a CTS packet to reserve the channel for the ZigBee node) or continue to make use of the channel (ignoring the request).

V. Cross-Technology Signaling

We describe next how a ZigBee node informs a Wi-Fi device about its channel requirements. As discussed in Sec. III-B, we cannot reuse existing CTC schemes, as they require a tight synchronization between devices, which would produce long delays neutralizing the benefits of our coordination scheme. We hence propose a cross-technology signaling scheme in which a ZigBee node quickly transmits control information to a Wi-Fi device. In this scheme, the Wi-Fi receiver does not need to fully decode ZigBee's signal. The control information can be obtained by only detecting the *existence* of a ZigBee transmission, a much more coarse-grained task. The Wi-Fi device, indeed, only needs to detect the change in the received CSI sequence with a sliding window, omitting the need of synchronization. Hence, the existence of the ZigBee transmission acts as *one-bit information* conveying the white space request of a nearby ZigBee node.

In the cross-technology signaling process, the ZigBee node tries to notify the Wi-Fi device about its need to access the channel by overlapping some control packets with Wi-Fi's subcarrier. Here, ZigBee's control packets are carefully customized to achieve reliable and efficient cross-technology

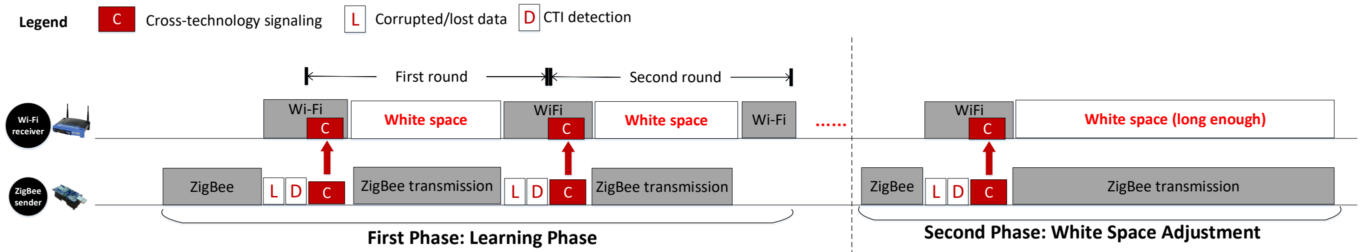


Fig. 4. The adaptive white space allocation in BiCord involves two phases: the learning phase and the white space adjustment phase.

signaling. Specifically, the control packets are long enough (set as 120 bytes in our implementation) to cover two continuous Wi-Fi packets. This makes sure that ZigBee's control packets always overlap with Wi-Fi packets. The transmission power of the ZigBee node is carefully selected to ensure that ZigBee's control packets can be reliably detected by the Wi-Fi receiver, as discussed in Sec. VII-A.

There is still a key challenge in the cross-technology signaling scheme, i.e., how to distinguish between ZigBee signal and noise, as they both create jitter in the CSI sequence. Fig. 3 (a) and (b) show the CSI sequence that is influenced by strong noise and ZigBee signal, respectively. We can see that these two sources of signals leave similar hints on the CSI sequence, making it easy to confuse the strong noise as a ZigBee transmission. In traditional CTC schemes, the confusion between ZigBee's signal and noise only incurs bit error, which can be further corrected. In our case, however, such a confusion leads to a high false positive rate in ZigBee detection, which would result in many unwanted white spaces.

A naïve solution to distinguish between ZigBee signal and noise consists in analyzing the amplitude of the jitter, as the jitter is usually low in the presence of noise. However, since strong noise occasionally occurs as well, we use multiple control packets to transmit the one-bit information, and leverage the *continuity* of the ZigBee signal to distinguish it from strong noise. Fig. 3 (c) and (d) show the CSI sequence that is influenced by two and three ZigBee packets, respectively. By comparing Fig. 3 (a), (c), and (d), we can see that strong noise occurs occasionally, whereas the ZigBee signal causes strong fluctuations more often. Based on this observation, we design an algorithm to detect ZigBee transmissions based on the time-domain feature of the signal.

Specifically, we first classify the CSI samples into two types based on their amplitude, i.e., slight jitter and high fluctuation. The classification is performed based on a threshold, as shown in Fig. 3. Then we select out the samples that are labeled as "high fluctuation". If we can find out N samples within a time period T (we set $N = 2$ and $T = 5ms$ in our implementation), which indicates that there is a continuous influence on the CSI sequence, the ZigBee signal is detected. Note that when the Wi-Fi receiver detects a series of high fluctuations, it does not need to exactly identify which of them are caused by the ZigBee signal. It just need to detect the *existence* of the ZigBee signal based on the continuity of the high fluctuation samples (i.e., the one-bit information).

Clearly, the detection process becomes more reliable when the ZigBee node sends more control packets. To this end, the ZigBee node keeps transmitting control packets until it receives an ACK from the intended ZigBee receiver (i.e., the Wi-Fi device generates a white space for ZigBee transmission), or until the control packets number exceeds a threshold (i.e., the Wi-Fi device ignores the request). Note that when ZigBee selects an appropriate power, ZigBee's signaling behavior has a minimal impact on the packet reception rate of Wi-Fi links, i.e., a decrease of 1-6% [9]. Hence, by performing cross-technology signaling, the Wi-Fi device receives the request from ZigBee node with minimal performance degradation.

VI. Adaptive White Space Allocation

We introduce next how BiCord determines the length of the white space based on *one-bit information*. BiCord achieves this leveraging the fact that the traffic pattern (i.e., the length of one transmission burst) of ZigBee is relatively stable [35]. Hence, it can determine the length of the required white space using a strategy in which the Wi-Fi device estimates the ZigBee traffic pattern based on several rounds of cross-technology signaling.

Specifically, the strategy involves two phases: the *learning phase* and the *white space adjustment phase*, as shown in Fig. 4. In the learning phase, the Wi-Fi device starts with a white space that is usually shorter than one burst of ZigBee data. To complete one transmission, the ZigBee node needs to send the request multiple times. During these rounds of requests, the Wi-Fi device is able to estimate ZigBee's traffic pattern and can adjust the length of the white space based on the estimation result.

Learning phase. Initially, a Wi-Fi device does not know the exact requirements of the ZigBee node. As an attempt, it starts by generating a short white space (30 or 40 ms in our implementation). Such a short white space may not be sufficient to transmit a long burst of ZigBee data containing N_{burst} packets. Hence, a ZigBee node may suffer from packet loss after the Wi-Fi device resumes its operations, as shown in Fig. 4. In this case, the ZigBee node will again perform the cross-technology signaling to notify the Wi-Fi device about its need to access the channel. Accordingly, the Wi-Fi device will generate another short white space of the same length as the initial one. Fig. 5 shows the detailed process of one round in BiCord's learning phase. Each round includes one white space T_w generated by the Wi-Fi device, whose length is determined according to Eq. (1):

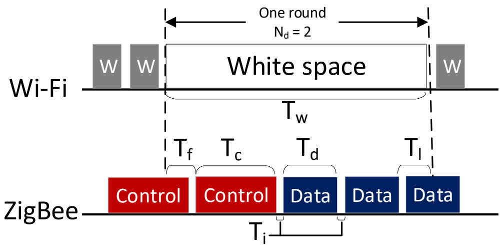


Fig. 5. The detailed process of one round in BiCord's learning phase.

$$T_w = T_f + T_c + T_d * N_d + T_i * N_d + T_l \tag{1}$$

T_c and T_d refer to the duration of a control and data packet, respectively, whereas T_i refers to the ZigBee's packet interval. N_d is the number of ZigBee data packets, whereas T_f refers to the time interval before the first control packet is sent (conversely, T_l is the time interval of the last data packet who could not be entirely sent within the white space). This procedure repeats for several rounds until the ZigBee node completes its transmissions. The number of rounds is given by the minimum N_{round} that meets $N_d * N_{round} \geq N_{burst}$. Considering a simple example where the ZigBee node has four packets to send and the white space generated by the Wi-Fi device accommodates two ZigBee data packets (as shown in Fig. 5), the Wi-Fi device need to spend two rounds to provide enough white space for the transmission of the entire burst of data. The end of ZigBee's transmissions is detected once the Wi-Fi device no longer detects ZigBee traffic for a given time after it resumes its transmissions (20 ms in our implementation). Once the ZigBee node completes its transmissions, the Wi-Fi device starts estimating the length of one ZigBee burst by calculating $T_{estimation} = (T_w - 2 * T_c) * N_{round}$. To avoid over-provisioning the white space length, the Wi-Fi device obtains a conservative estimation by subtracting $2 * T_c$ for each round. The estimation error is given by $T_{less} = T_c + T_d * N_{burst} + T_i * N_{burst} - T_{estimation}$. The learning phase iterates for several times before $T_{estimation}$ just covers one ZigBee burst. After that, the Wi-Fi device always generates a white space that is long enough for ZigBee transmissions when detecting a cross-technology signal.

White space adjustment. If the traffic pattern of a Zigbee node does not change, the Wi-Fi device can keep generating a white space that exactly fits its requirements. However, in case the transmission pattern of the ZigBee node changes or if there are multiple ZigBee nodes with different traffic pattern coexisting in the surroundings, the generated white space length needs to be re-adjusted. To this end, BiCord makes use of a traffic pattern re-estimation mechanism that can be triggered periodically or once a variation in the traffic pattern is detected at runtime. Specifically, when the length of one ZigBee burst becomes longer, the ZigBee node will again suffer packet corruption or loss and send a cross-technology signal: this triggers the re-estimation process. If the length of one ZigBee burst becomes shorter, this will not be noticed by the Wi-Fi device, which will keep generating a white space longer than required (leading to a low channel utilization). To avoid this, we set an expiring timer (set to 10 s in our implementation) to trigger a periodic re-estimation of ZigBee's traffic pattern. Note that ZigBee traffic can also be aperiodic, as even aperiodic transmissions trigger a cross-technology signal and the consequent generation of a white space.

VII. Implementation Details

We implement BiCord on commercial off-the-shelf Wi-Fi devices (Intel 5300 series) and ZigBee nodes (TelosB motes running Contiki 3.0). In this section we provide additional details and remarks on BiCord's implementation.

A. CTI detection

When a ZigBee node suffers from data corruption or a low packet reception rate, it first performs CTI detection. The goal of CTI detection is two-fold. First, it determines whether Wi-Fi traffic is the root cause of the unreliable transmissions. Second, if the interference does come from a Wi-Fi device, it autonomously determines the transmission power to be used in the cross-technology signaling module. Since the transmission power should be selected according to the features of the Wi-Fi transmitter (e.g., the Wi-Fi device's transmission power and the distance between the Wi-Fi transmitter and the ZigBee node), the ZigBee node negotiates an appropriate transmission power with each Wi-Fi device in advance using the method proposed in [9]. After that, the ZigBee node constructs a PowerMap containing the power to be used when the channel is occupied by a Wi-Fi device. Hence, once a ZigBee node detects Wi-Fi traffic, it identifies the device it belongs to and sets the transmission power as the corresponding value stored in the PowerMap.

Discerning Wi-Fi traffic from other interference. A ZigBee node needs to first understand whether the ongoing traffic comes from a Wi-Fi sender. This can be achieved using the method proposed in ZiSense [21], which shows that the RSSI trace of different technologies exhibit different physical layer features. To this end, we use four features: average on-air time, minimum packet interval, peak to average power ratio (i.e., the ratio of maximum and average RSSI sample), as well as under noise floor (i.e., an indicator of whether an RSSI lower than the noise floor is detected). Once channel activity is detected, the ZigBee node extracts a segment of the RSSI sequence, calculates the aforementioned features, and feeds them to a decision tree model to infer whether the channel activity comes from a nearby Wi-Fi device. If this is the case, the ZigBee node needs to identify the exact Wi-Fi transmitter from which it originates. To this end, some more fine-grained features are extracted from the RSSI sequence, as illustrated in Smoggy-Link [20]. Those features, which include energy span, energy level, energy variance, and occupancy level, form a fingerprint for each device. We exploit the *k-means* clustering technique to discriminate different devices based on the Manhattan distance between their fingerprints. If the detected channel activity is not coming from a nearby Wi-Fi device (e.g., it comes from other interference sources such as Bluetooth and microwave ovens), the ZigBee node does not perform cross-technology signaling and directly returns to sleep mode. Note that BiCord is orthogonal to existing interference recovery mechanisms such as forward error correction [15], and can be hence integrated into those mechanisms to further improve reliability.

Accuracy of CTI detection. To verify the reliability of CTI detection, we use a ZigBee node (collector) to gather RSSI segments of various devices and then conduct interference identification. To collect ZigBee RSSI segments, we instruct a ZigBee sender to broadcast packets of 50 bytes every 2 ms. To collect Bluetooth RSSI segments, we establish a link between a Bluetooth headset and a computer playing music nearby the ZigBee collector. As for Wi-Fi, we place a Wi-Fi sender periodically broadcasting 100-byte packets every 1 ms at a distance of 1, 3, and 5 meters from the ZigBee collector. For every setting, the collector records the RSSI sequence 200 times, at a frequency of 40 kHz for 5 ms. The average accuracy in detecting Wi-Fi traffic among RSSI segments from all technologies is 96.39%. As for different Wi-Fi devices, the ZigBee node achieves an average identification accuracy of 89.76%, with a standard deviation of 2.14%. These experimental results show the ability of a ZigBee node to identify each Wi-Fi device with high accuracy.

B. Energy cost of BiCord on ZigBee nodes

To coordinate with Wi-Fi devices and obtain channel resources, ZigBee nodes need to detect Wi-Fi traffic and send control packets to directly transfer information. By selecting an appropriate transmission power, a ZigBee node usually needs to send only one or two control packets to notify the nearby Wi-Fi device. Furthermore, with the help of the adaptive white space allocation module described in Sec. VI, ZigBee nodes only performs cross-technology signaling once, as Wi-Fi devices learn the duration of ZigBee transmissions.

Considering a ZigBee node sending ten packets of 120 bytes in a burst under strong Wi-Fi interference, BiCord generally costs 10% to 21% extra energy in comparison to sending these data packets in a clear channel. However, in a noisy environment, the ZigBee has to retransmit packets corrupted by interference. The energy consumption of BiCord is less than the cases where ZigBee nodes need to re-transmit more than two packets. Moreover, in traditional approaches, ZigBee needs keep sensing the channel to analyze the channel hints or passively wait for Wi-Fi's notification, which inevitably leads to long delays and even higher energy costs. Note that an option to shrink the energy cost related to the transmission of BiCord's control packets is the ability to reuse them for data transmission as well: we will investigate this in future work.

C. Impact of BiCord on Wi-Fi operations

To implement BiCord on commercial Wi-Fi devices, the CSI extractor tool was installed and configured to collect CSI readings at a rate of 2 kHz. Note that the CSI collection feature is an integral part of standards such as 802.11ac, and can be installed in modern Wi-Fi chipsets. The cross-technology signaling module has little effect on ongoing Wi-Fi traffic, as we noticed at most a decrease of 1–6% of the device's packet reception rate [9]. The adaptive white space allocation module has a controllable effect on the performance of a Wi-Fi device. As discussed in Sec. IV, Wi-Fi devices can decide whether to defer their transmissions based on their own application requirements. For example, when having high-priority traffic (e.g., video streaming), a Wi-Fi device can ignore ZigBee signals and keep transmitting packets to minimize its delay. In case its traffic is delay-insensitive (e.g., file transfer), a Wi-Fi device can make space for ZigBee transmissions. Therefore, BiCord improves coexistence and maximizes ZigBee performance with a configurable impact on Wi-Fi's performance.

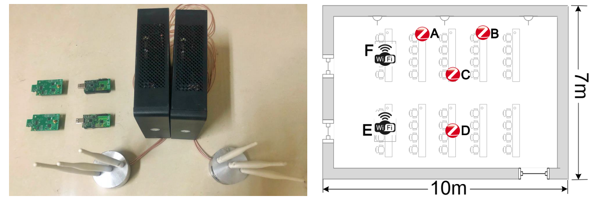


Fig. 6. Experimental setup used in our evaluation.

D. Extension to other coexistence scenarios

Although we have tailored the design and implementation of BiCord for Wi-Fi and ZigBee devices, the idea of directly-coordinated channel allocation can be extended to different coexistence scenarios. For example, using CTC from ZigBee to Bluetooth [31], BiCord is able to coordinate coexisting ZigBee and Bluetooth networks. With the emergence of new CTC schemes enabling communication among different wireless technologies, we believe that the concepts behind BiCord can be applied in different contexts and help maximizing the coexistence among several wireless systems.

VIII. EVALUATION

We evaluate the performance of BiCord experimentally. After describing our experimental setup (Sec. VIII-A), we evaluate the cross-technology signaling (Sec. VIII-B) and adaptive white space allocation (Sec. VIII-C) modules. We then compare the performance of BiCord to that of ECC (Sec. VIII-D) and analyze the impact of various parameters on BiCord's performance (Sec. VIII-E). We finally study the performance of BiCord in mobile scenarios (Sec. VIII-F) and in the presence of prioritized Wi-Fi traffic (Sec. VIII-G).

A. Experimental setup

All our experiments are carried out in office environments with background noise. Fig. 6 shows the location of Wi-Fi sender (E) and receiver (F), at a distance of 3 m from each other. We place the ZigBee sender at four different locations (A–D) and lay down the ZigBee receiver 1–5 m away from it. The Wi-Fi sender transmits packets of 100 bytes with a 1 ms interval. The ZigBee sender uses a transmission power of −7 dBm and suffers a packet loss of over 95% when the nearby Wi-Fi sender is transmitting data.

If not specified, we let Wi-Fi devices make space for ZigBee traffic every time they detect a ZigBee signal. We further configure the Wi-Fi devices to operate on channel 11 or 13, and ZigBee nodes to transmit on IEEE 802.15.4 channel 24 or 26, such that they overlap in the frequency domain.

B. Performance of cross-technology signaling

Setup. We let a ZigBee node send every 16 ms a burst of control packets, each with a length of 120 bytes, for 600 times. We vary power level, number of control packets, and location of ZigBee sender. The Wi-Fi receiver runs CSI analysis and outputs positive result when detecting ZigBee traffic. We measure the *precision* and *recall* of the detection. The precision measures the ratio of true positive results produced by ZigBee's signaling to the number of positive results outputted by the Wi-Fi device. The recall is the ratio of ZigBee's signal that makes a Wi-Fi device output a positive result.

Results. Table I and II show the corresponding results. In general, with an appropriate power, ZigBee nodes sending 4 control packets can achieve a precision of 93.6%, 90.6%, 86.4%, 86.4% and a recall of 93.5%, 89.6%, 92%, 78%, respectively, at location A, B, C, and D. Independently of the location and the power used by the ZigBee node, both the precision and the recall of cross-technology signaling increase with the number of control packets sent by the ZigBee node.

The power and location of the ZigBee nodes affect the performance as well. In particular, at location A or B, the precision and recall gradually decrease when the ZigBee node lowers its transmission power. As for location C, we observe that when sending packets at -1 dBm, the precision and recall is the highest. At this location, if the ZigBee sender uses a higher power, it may cause the Wi-Fi device to back-off when conducting CCA, and may fail in cross-technology signaling. At location D, the ZigBee sender is closer to the Wi-Fi sender and adopts a lower power to prevent Wi-Fi to back-off. This emphasize the need to select an appropriate transmission power to maximize BiCord's performance, using the methods in Sec. VII-A.

C. Performance of adaptive white space allocation

Setup. We evaluate the performance of BiCord's adaptive white space allocation module by letting a Wi-Fi device estimate the duration of ZigBee's transmissions and adapt the length of the white space. In particular, a ZigBee node sends bursts of 5, 10, or 15 packets, with a length of 50 bytes each, every 200 ms. The Wi-Fi device uses an initial white space of 30 ms or 40 ms during the learning phase, and sets the duration of ZigBee control packets as 8 ms during estimation. We carry out 30 experiments with the ZigBee node at location A and B, respectively, calculating the average number of iterations and length of the provided white space.

Results. Fig. 7 shows the process of adaptive white space allocation by measuring the length of the white space provided by the Wi-Fi device during the adjustment phase. The ZigBee node sends 10 packets in a burst and the Wi-Fi device uses steps of 30 ms. The Wi-Fi device keeps learning the duration of ZigBee transmissions and lengthens the white space. After about 5 iterations, the Wi-Fi device converges and reserves a white space of about 70 ms, which is sufficient to contain the burst of ZigBee data that lasts 62.7 ms.

Fig. 8 shows the number of iterations needed by the Wi-Fi device to adjust the white space – in average always below 8. When a ZigBee node sends more packets in a burst or when the Wi-Fi device uses a shorter step length, the number of iterations increases. The (relatively) worse performance at location A can be explained by the proximity of the ZigBee node to the Wi-Fi device, which causes the latter to interpret ZigBee data packets as a channel request. Consequently, the ZigBee node does not need to send control packets that occupy the white space left by Wi-Fi, leading to Wi-Fi's underestimation of the duration of ZigBee's data traffic.

Fig. 9 shows the white space generated after the adjustment phase. The Wi-Fi device is able to lengthen the white space when the duration of ZigBee transmissions increases. When the Wi-Fi device adopts a longer step size, it tends to leave longer white spaces to cover ZigBee transmissions. When calculating the requirements of ZigBee traffic, we find that the generated white space is larger than needed by 27.1%, 12.5%, and 20.4% when ZigBee transmits $5, 10$, and 15 packets in each round – an acceptable compromise considering that, in contrast to ECC, the white space will surely be used.

D. Comparison with ECC

We evaluate the overall performance of BiCord and compare it with ECC in terms of (i) channel utilization, (ii) ZigBee's transmission delay, as well as (iii) throughput. To calculate the channel utilization, we measure the transmission time of both Wi-Fi and ZigBee devices and add them together.

Setup. We let a ZigBee node send bursts of 5 packets with a payload of 50 bytes each and an average interval between two bursts of 101.56 ms (13 ticks), 203.12 ms (26 ticks), 406.24 ms (52 ticks), 1s (128 ticks) and 2s (256 ticks), to represent different intensities of traffic. To guarantee a reliable transmission, every ZigBee packet should be acknowledged by its intended recipient. We assume that the data traffic of ZigBee nodes is originated following a Poisson process. The above setting reflects the conventional practice in real-world ZigBee implementations [35]. As ECC cannot predict the evolution of ZigBee traffic using the folding technique, it periodically generates white spaces of fixed length. We fix this period to 100 ms, the typical setting in ECC, and set the length of the white space as 20 ms, 30 ms, and 40 ms to cover most of ECC's settings. We make use of a ZigBee node at location A transmitting 1000 packets at 0 dBm.

Channel utilization. Fig. 10(a) shows the channel utilization achieved by BiCord and ECC. Overall, BiCord's channel utilization in Fig. 10(a) is always larger than 80%, independently of the average interval between two ZigBee bursts. BiCord especially outperforms ECC with infrequent ZigBee traffic: with an interval between two bursts of 2 s, *BiCord achieves*

TABLE I
THE PRECISION OF CROSS-TECHNOLOGY SIGNALING AT DIFFERENT LOCATION WITH DIFFERENT PARAMETERS.

Power (dBm)	0			-1			-3		
Packet Number	3	4	5	3	4	5	3	4	5
Location A	0.8548	0.9355	0.95	0.8533	0.93	0.9714	0.8286	0.9365	0.9525
Location B	0.8571	0.9057	0.9649	0.8	0.8333	0.9	0.7183	0.8571	0.9167
Location C	0.5862	0.7333	0.8	0.83	0.8636	0.9	0.72	0.8222	0.86
Location D	0.6125	0.71	0.73	0.7222	0.76	0.83	0.8	0.8636	0.91

TABLE II
THE RECALL OF CROSS-TECHNOLOGY SIGNALING AT DIFFERENT LOCATION WITH DIFFERENT PARAMETERS.

Power (dBm)	0			-1			-3		
Packet Number	3	4	5	3	4	5	3	4	5
Location A	0.88	0.9355	0.9828	0.8889	0.9538	0.9839	0.9155	0.9219	0.9825
Location B	0.7273	0.8955	0.8302	0.7727	0.8421	0.9483	0.62	0.7969	0.8182
Location C	0.73	0.7526	0.762	0.87	0.92	0.9	0.68	0.675	0.75
Location D	0.68	0.6383	0.67	0.63	0.7029	0.71	0.7358	0.78	0.82

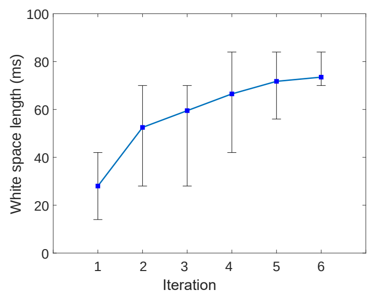


Fig. 7. Adjustment process.

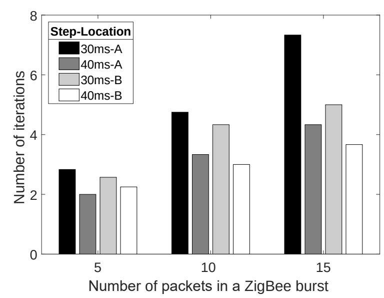


Fig. 8. No. of iterations needed.

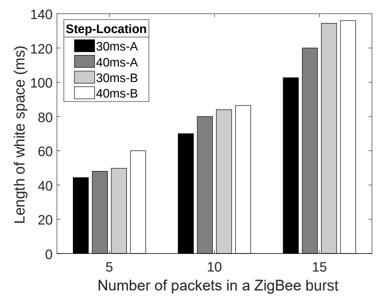
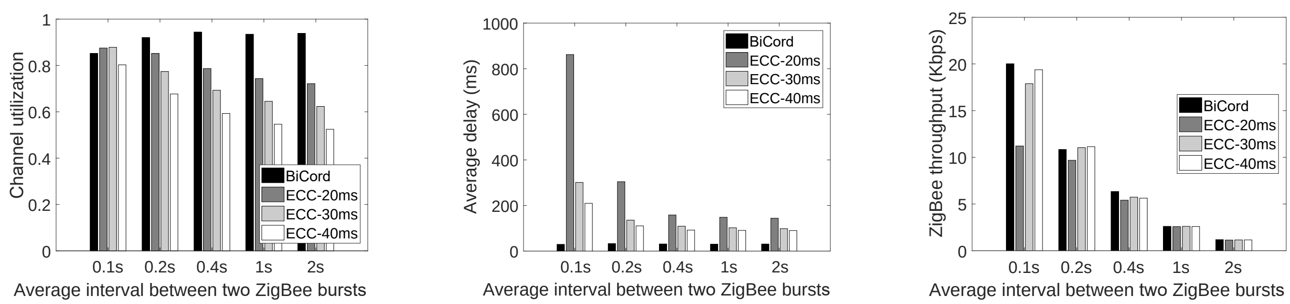


Fig. 9. White space length.



(a) Channel utilization
(b) ZigBee's transmission delay
(c) ZigBee throughput

Fig. 10. Comparison between ECC and BiCord on channel utilization (a), transmission delay (b), and throughput (c).

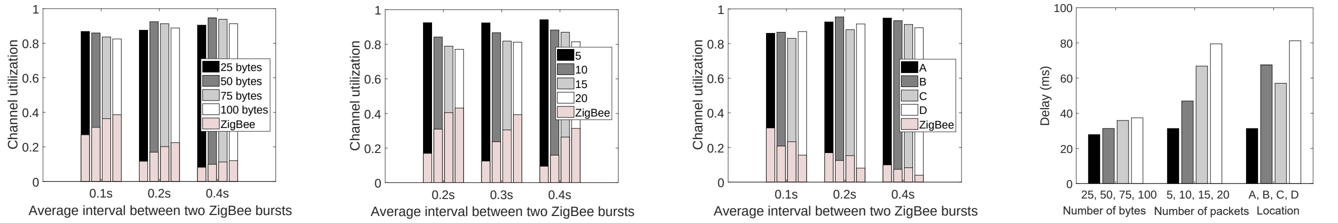
a 50.6% higher channel utilization than ECC. The utilization of ECC largely depends on the frequency of ZigBee transmissions: if packets are sparsely distributed in the channel, the Wi-Fi device over-provisions the white space, resulting in a large waste of resources. This is not the case with BiCord.

ZigBee's transmission delay. As shown by Fig. 10(b), in BiCord, the average ZigBee delay is well-below 30 ms regardless of Wi-Fi traffic, thanks to the coordination enabled by cross-technology signaling. As for ECC, the ZigBee delay is directly related to the intensity of Wi-Fi and ZigBee traffic. When Wi-Fi traffic is too frequent to provide enough white space for ZigBee, the ZigBee transmissions incur a long delay. This is because when the white space left by the Wi-Fi device is not long enough to accommodate one burst of ZigBee data, the ZigBee node has to defer its transmissions until the Wi-Fi device leaves a new white space. Hence, in terms of delay, BiCord outperforms ECC in average by 84.2% for the scenarios shown in Fig. 10(b).

ZigBee's throughput. As shown in Fig. 10(c), BiCord can always provide the white space length that is required by ZigBee nodes, maximizing their throughput. In contrast, ECC can only provide a fixed white space length, so the throughput of ZigBee nodes is limited by the amount of packets that can fit within such a white space.

E. Impact of Different BiCord's Parameters

Setup. We now study in detail the impact of a few parameters such as the length and the number of ZigBee's packets, as well as the location of the ZigBee sender. We vary one parameter at a time, measure the channel utilization and average delay



(a) Length of ZigBee packets (b) Number of ZigBee packets (c) Location of ZigBee sender (d) Average delay of ZigBee packets

Fig. 11. Performance of BiCord as a function of different parameters.

of ZigBee packets, and show our results in Fig. 11. To show more in detail the allocation of the channel, we use a different color (pink) to show the channel utilization of ZigBee, while the remaining color represents Wi-Fi's utilization. By default, we use a ZigBee sender at location A transmitting a burst of 5 packets of 50 bytes at different intervals.

Results. As shown in Fig. 11(a) and 11(b), when the duration of a ZigBee burst increases due to a larger packet length or to a higher number of packets within a burst, ZigBee's channel occupation increases accordingly. Overall, the total channel utilization achieved by BiCord remains in the order of 80%, independently of these two parameters, confirming the results previously presented in Fig. 10(a).

Fig. 11(c) shows the channel utilization as a function of the location of the ZigBee sender. The performance of BiCord at different locations is proportional to the effectiveness of the cross-technology signaling module. As discussed in Sec. VIII-B, the ZigBee sender at location B is far away from the ZigBee receiver, so it is more difficult for ZigBee to correctly signal its presence to the Wi-Fi device even when using a transmission power of 0 dBm. As a result, the ZigBee channel allocation is highest at location A and C, which confirms the results in Table I and II[3]. Also in this case, the total channel utilization achieved by BiCord remains in the order of 80%, independently of the location of ZigBee nodes.

Fig. 11(d) shows the average delay of every ZigBee packet. This value increases with the duration of the burst, because the later packets have to wait for the former packets to be sent. At different locations, ZigBee experiences a lower delay when the cross-technology signaling is more effective. Based on our experiments, the delay is kept under 80 ms, and is approximately 30 ms if the ZigBee sender transmits a few packets with fewer length.

F. Performance in Mobile Scenarios

Setup. We also evaluate BiCord in two mobile scenarios. In a first set of experiments, we let one person walk around the Wi-Fi receiver and ZigBee sender at a speed of 1–2 m/s. In a second set of experiments, we move the ZigBee sender within a distance of 1 m. We keep all other settings for the ZigBee node identical to those used in Sec. VIII-D.

Results. We compare the channel utilization of the two mobile scenarios with that of a static scenario. Fig. 12 shows our

[3]Note that BiCord makes use of a transmission power of 0 dBm, 0 dBm, -1 dBm and -3 dBm at locations A, B, C and D, respectively.

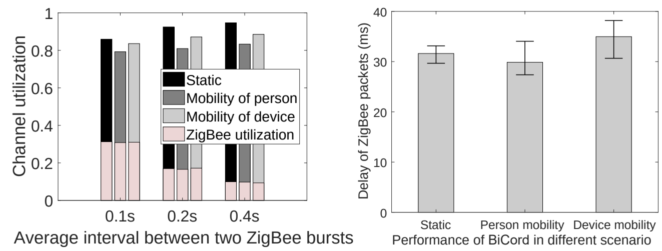


Fig. 12. Channel utilization and average delay of ZigBee packets in mobile scenarios.

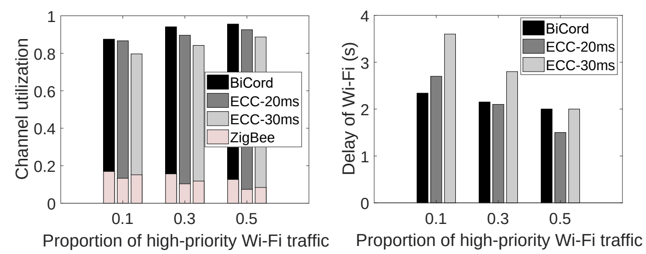


Fig. 13. Channel utilization and average Wi-Fi delay considering different priorities of Wi-Fi traffic.

results: as expected, the utilization is slightly lower (at most 9%) than in a static scenario, but BiCord's performance is not significantly degraded. In the first scenario, the walking person influences the CSI values of Wi-Fi packets, making the Wi-Fi device misjudge the presence of ZigBee traffic in a few cases. As a result, some of the generated white spaces generated remain unused, leading to a slightly lower utilization. In the second scenario, the ZigBee sender may suffer additional data corruption or loss and needs to re-transmit data packets. The longer transmission makes the Wi-Fi device leave more white space, decreasing channel utilization by 4.6%.

Fig. 12 also shows the delay of ZigBee packets. In the first scenario, the average delay of ZigBee packets is lower because the Wi-Fi device may generate a white space before ZigBee's transmissions. In the second scenario, instead, the ZigBee node needs to send more control packets before sending data, which slightly raises the delay (by 3.13 ms in average).

G. Prioritization of Wi-Fi Traffic

Setup. In this experiment, we let the Wi-Fi device have both high-priority traffic (video streaming) and low-priority traffic (file transfer). As for high-priority traffic, the Wi-Fi device does not respond to the ZigBee's request. The whole Wi-Fi traffic lasts for 10 s, where the proportion of the high-priority traffic is adjusted from 0.1 to 0.5. The ZigBee node generates bursts of five 50-bytes packets with the average interval of 200 ms. We measure the total channel utilization and the ZigBee utilization, and show the results in Fig. 13 (left).

Results. The channel utilization of BiCord outperforms ECC-20ms and ECC-30ms by 3.11% and 9.76%, respectively. In terms of ZigBee utilization, BiCord outperforms ECC-20ms and ECC-30ms by 46.05% and 27.97%, respectively.

As Wi-Fi devices ignore the request from ZigBee ndoes when having high-priority traffic, the high-priority Wi-Fi transmissions face nearly zero delay. Specifically, Fig. 13 (right) shows the delay of low-priority Wi-Fi traffic. In average, BiCord reduces the delay by 6%, compared to ECC. In ECC, the Wi-Fi device experience less delay only with shorter white space and less low-priority traffic. In this case, ECC leaves less white space and ZigBee suffers very high delays (up to 300 ms). The delay experienced by ZigBee nodes is similar to that shown in Fig. 10(b), so we omit this figure to fit the paper within the space constraints.

IX. Conclusions

This work introduces BiCord, a novel channel coordination scheme enabling a bidirectional interaction between constrained wireless devices (ZigBee) and more powerful appliances (Wi-Fi) operating in the same frequency band. To this end, we introduce a cross-technology signaling method in which ZigBee nodes efficiently inform nearby Wi-Fi devices about their need to access the channel. We also design an adaptive white space allocation scheme for Wi-Fi devices to learn and meet the need of ZigBee nodes, which facilitates an efficient cross-technology channel coordination. Real-world experiments on commercial Wi-Fi and ZigBee devices under various scenarios demonstrate the superior performance of BiCord compared to state-of-the-art approaches.

References

[1] Gang Zhou, John A. Stankovic and Sang H. Son, "Crowded Spectrum in Wireless Sensor Networks", in Proc. IEEE EmNets Workshop, 2006.

[2] Carlo Alberto Boano, Thiemo Voigt, Claro Noda, Kay Römer and Marco Antonio Zúñiga, "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation", in Proc. IEEE IPSN, 2011.

[3] Kaushik R. Chowdhury and Ian F. Akyildiz, "Interferer Classification, Channel Selection and Transmission Adaptation for Wireless Sensor Networks", in Proc. IEEE ICC, 2009.

[4] Gang Jun Huang, Guoliang Xing, Gang Zhou and Ruogu Zhou, "Beyond Co-existence: Exploiting WiFi White Space for ZigBee Performance Assurance", in Proc. IEEE ICNP, 2010.

[5] Zhijun Li and Tian He, "WEBee: Physical-Layer Cross-Technology Communication via Emulation", in Proc. ACM MobiCom, 2017.

[6] Zhimeng Yin, Zhijun Li, Song Min Kim and Tian He, "Explicit Channel Coordination via Cross-technology Communication", in Proc. ACM MobiSys, 2018.

[7] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Zihao Yu and Yunhao Liu, "LEGO-Fi: Transmitter-Transparent CTC with Cross-Demapping", in Proc. IEEE INFOCOM, 2019.

[8] Shuai Wang, Song Min Kim and Tian He, "Symbol-level Cross-technology Communication via Payload Encoding", in Proc. IEEE ICDCS, 2018.

[9] Xiuzhen Guo, Yuan He, Xiaolong Zheng, Liangcheng Yu and Omprakash Gnawali, "ZigFi: Harnessing Channel State Information for Cross-Technology Communication", in Proc. IEEE INFOCOM, 2018.

[10] Weiguo Wang, Xiaolong Zheng, Yuan He and Xiuzhen Guo, "AdaComm: Tracing Channel Dynamics for Reliable Cross-Technology Communication", in Proc. IEEE SECON, 2019.

[11] J.J. Garcia-Luna-Aceves, "Carrier-tone multiple access with collision avoidance and detection", in Proc. IEEE WCNC, 2018.

[12] Tianji Li, Mi Kyung Han, Apurv Bhartia, Lili Qiu, Eric Rozner, Yin Zhang and Brad Zarikoff, "CRMA: Collision-Resistant Multiple Access", in Proc. ACM MobiCom, 2011.

[13] Kai-Cheng Hsu, Kate Ching-Ju Lin and Hung-Yu Wei, "Inter-client interference cancellation for full-duplex networks", in Proc. IEEE INFOCOM, 2017.

[14] Xinyu Zhang and Kang G. Shin, "Enabling Coexistence of Heterogeneous Wireless Systems: Case for ZigBee and WiFi", in Proc. ACM MobiHoc, 2011.

[15] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu and Andreas Terzis, "Surviving Wi-Fi Interference in Low Power ZigBee Networks", in Proc. ACM SenSys, 2010.

[16] Christos Gkantsidis, Wenjun Hu, Peter Key, Bozidar Radunovic, Pablo Rodriguez and Steluta Gheorghiu, "Multipath Code Casting for Wireless Mesh Networks", in Proc. ACM CoNEXT, 2007.

[17] Dimitrios Koutsonikolas, Y. Charlie Hu and Chih-Chun Wang, "Pacifier: High-Throughput, Reliable Multicast without "Crying Babies" in Wireless Mesh Networks", in Proc. IEEE INFOCOM, 2009.

[18] Sangki Yun and Lili Qiu, "Supporting WiFi and LTE co-existence", in Proc. IEEE INFOCOM, 2015.

[19] Micah Z. Brodsky and Robert T. Morris, "In defense of Wireless Carrier Sense", in Proc. ACM SIGCOMM, 2009.

[20] Meng Jin, Yuan He, Xiaolong Zheng, Dingyi Fang, Dan Xu, Tianzhang Xing and Xiaojiang Chen, "Smoggy-Link: Fingerprinting Interference for Predictable Wireless Concurrency", in Proc. IEEE ICNP, 2016.

[21] Xiaolong Zheng, Zhichao Cao, Jiliang Wang, Yuan He and Yunhao Liu, "ZiSense: Towards Interference Resilient Duty Cycling in Wireless Sensor Networks", in Proc. ACM SenSys, 2014.

[22] Anwar Hithnawi, Hossein Shafagh and Simon Duquennoy, "TIIM: Technology-Independent Interference Mitigation for Low-power Wireless Networks", in Proc. ACM IPSN, 2015.

[23] Song Min Kim and Tian He, "FreeBee: Cross-Technology Communication via Free Side-Channel", in Proc. ACM MobiCom, 2015.

[24] Xiuzhen Guo, Xiaolong Zheng and Yuan He, "Wizig: Cross-technology Energy Communication over a Noisy Channel", in Proc. IEEE INFOCOM, 2017.

[25] Kameswari Chebrolu and Ashutosh Dhekne, "Esense: communication through energy sensing", in Proc. ACM MobiCom, 2009.

[26] Xiaolong Zheng, Yuan He and Xiuzhen Guo, "StripComm: Interference-Resilient Cross-Technology Communication in Coexisting Environments", in Proc. IEEE INFOCOM, 2018.

[27] Zhimeng Yin, Wenchao Jiang, Song Min Kim and Tian He, "C-Morse: Cross-technology communication with transparent morse coding", in Proc. IEEE INFOCOM, 2017.

[28] Rainer Hofmann, Carlo Alberto Boano and Kay Römer, "X-Burst: Enabling Multi-Platform Cross-Technology Communication between Constrained IoT Devices", in Proc. IEEE SECON, 2019.

[29] Xiuzhen Guo, Yuan He, Jia Zhang and Haotian Jiang, "WIDE: Physical-level CTC via Digital Emulation", in Proc. IEEE IPSN, 2019.

[30] Yongrui Chen, Zhijun Li and Tian He, "TwinBee: Relable Physical-layer Cross-technology Communication with Symbol-level coding", in Proc. IEEE INFOCOM, 2018.

[31] Wenchao Jiang, Song Min Kim, Zhijun Li and Tian He, "Achieving Receiver-Side Cross-Technology Communication with Cross-Decoding", in Proc. ACM MobiCom, 2018.

[32] Zhijun Li, Yan Li, Wenchao Jiang and Tian He, "BlueBee: Physical-Layer Cross-Technology Communication via Emulation", in Proc. ACM SenSys, 2017.

[33] Ruofeng Liu, Zhimeng Yin, Wenchao Jiang and Tian He, "LTE2B: Time-Domain Cross-Technology Emulation under LTE Constraints", in Proc. ACM SenSys, 2019.

[34] Wei Wang, Shiyue He, Liang Sun, Tao Jiang and Qian Zhang, "Cross-Technology Communications for Heterogeneous IoT Devices Through Artificial Doppler Shifts", IEEE Trans. Wireless Commun., vol.18, no.2, 2019.

[35] Yunhao Liu, Yuan He, Mo Li, Jiliang Wang, Kebin Liu, Lufeng Mo, Wei Dong, Zheng Yang, Min Xi and Jizhong Zhao, "Does wireless sensor network scale? A measurement study on GreenOrbs", IEEE Transactions on Parallel & Distributed Systems,vol. 24, pp.1983-1993, 2013.