X-MIMO: Cross-Technology Multi-User MIMO

Shuai Wang[§†], Woojae Jeong[†], Jinhwan Jung[†], and Song Min Kim[†*]
[†]Korea Advanced Institute of Science and Technology (KAIST)
[§]George Mason University

ABSTRACT

Multi-user MIMO (MU-MIMO) is a widely-known, fundamental technique to significantly improve the spectrum efficiency. While there is a great demand for spectrum efficiency and massive scalability under explosively increasing IoT, hardware limitations make it particularly challenging for the mechanism to be transferred to the IoT (e.g., ZigBee) domain. This paper presents X-MIMO, a zero-cost, software-only cross-technology MU-MIMO for commodity ZigBee. As the first work to shed the light on the feasibility of MU-MIMO on commodity IoT, X-MIMO leverages on cross-technology communication (CTC) to turn the pervasively-deployed WiFi AP into MU-MIMO transmitter, delivering different packets to multiple ZigBees in parallel. X-MIMO uniquely exploits WiFi CSI to extract the accurate physical layer signal of the ZigBee packet and the WiFi-ZigBee channel coefficient. Rigorous derivation shows that X-MIMO's precoding is inherently immune to the uncertainties of the commodity devices, making X-MIMO highly reliable in practice. Lastly, spectrum-efficient emulation is proposed to maximize the spectrum reuse. We implement and comprehensively evaluate the performance of X-MIMO on commodity devices (Atheros AR9334 WiFi NIC and TelosB CC2420) as well as on USRP B210 for in-depth analysis. Results reveal that X-MIMO achieves 495 Kbps with <1% symbol error rate (SER) and 704.24 Kbps with 6.1% SER for two and three streams, respectively. Near-linear increase of the throughput effectively demonstrates the feasibility of X-MIMO.

CCS CONCEPTS

• **Networks** → **Wireless local area networks**.

KEYWORDS

MU-MIMO; Cross-tech. Communication; Wireless Communication

ACM Reference Format:
Shuai Wang, Woojae Jeong, Jinhwan Jung, and Song Min Kim. 2020. X-MIMO: Cross-Technology Multi-User MIMO. In *Proceedings of The 18th ACM Conference on Embedded Networked Sensor Systems (SenSys '20), Nov 16–19, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3384419.3430723

*Song Min Kim is the corresponding author (songmin@kaist.ac.kr).
†Shuai Wang was a visiting student in KAIST, officially affiliated with GMU.

SenSys '20, November 16–19, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-7590-0/20/11…$15.00
https://doi.org/10.1145/3384419.3430723

1 INTRODUCTION

The body of wireless devices is experiencing rapid growth with the emergence of the Internet of Things (IoT) era. The number of IoT devices is expected to grow as large as a trillion by 2035 [46], with the vision of providing pervasive services spanning every corner of our daily lives. To achieve this, the key factor in IoT is the capability to extend to an extreme scale in a spectrum efficient manner, thereby enabling prevalent deployment. This is indeed critical considering that the IoT standards inevitably suffer from a slow transmission rate (and thus low spectrum efficiency), in order to simplify the modulation and keep the receiver radio architecture simple, low-cost, and power-efficient. For instance, ZigBee and Bluetooth have 0.125 and 1 bits/s/Hz, which are 240 and 30 times lower spectrum efficiencies compared to WiFi 802.11n (30 bits/s/Hz).

MU-MIMO, by enabling a transmitter to simultaneously deliver different packets to multiple receivers, has been adopted in a wide range of practical wireless systems including WiFi and LTE. MU-MIMO serves as the foundational component to extend the scalability under limited channel resource, recently to a massive level (e.g., massive MIMO in 5G [33]) – a feature which IoT would critically benefit from. However, achieving this in the IoT domain is challenging due to the following intrinsic limitations: (i) Most of IoT devices are equipped with a single antenna, while MU-MIMO needs a multi-antenna transmitter. (ii) While channel estimation is an essential part of MU-MIMO, it is typically unavailable in IoT. This is because, for low-power operation and economical hardware, IoT devices are commonly designed as non-coherent receivers where channel estimation is not performed. To overcome these two practical limitations, the existing approaches rely on the high-end software-defined radio [41, 54] and complex signal processing [13, 19], unachievable with commodity IoT.

This paper presents X-MIMO, the first work to bring MU-MIMO into the picture of commodity IoT networking. X-MIMO is a zero-cost, software-only solution that uses pervasively-deployed commodity WiFi APs as the IoT MU-MIMO transmitter, to simultaneously deliver different packets to multiple ZigBee devices. X-MIMO does not require additional hardware or modification of firmware or driver. X-MIMO is inspired by the recent advancement in cross-technology communication (CTC), enabling commodity WiFi to transmit ZigBee packets via physical-layer signal emulation [36]. To uniquely enable MU-MIMO CTC, X-MIMO effectively leverages the MIMO capability of 802.11n (the most widely deployed WiFi variant) WiFi APs – That is, multiple antennas and multi-stream signal processing (typically 3 or above) for parallel transmission. Furthermore, among various MIMO technologies, unique features of MU-MIMO make it especially well-suited for the IoT scenario: First, MU-MIMO is designed to support receivers with a single antenna (or fewer than that of the sender) [37, 41, 47], which applies to most of the commodity IoT. Also, by using the technique of

precoding, what reaches the receiver (i.e., commodity IoT) is a legitimate ZigBee packet, thereby incurring no extra signal processing overhead on the (typically low-end) IoT.

X-MIMO is built with three new mechanisms of (i) cross-technology channel estimation, (ii) cross-technology precoding, and (iii) multi-stream CTC, where making them fully compatible with commodity devices incurs significant challenge. Cross-technology channel between ZigBee and WiFi is measured using WiFi CSI, from which the *physical-layer* signal (and the channel accordingly) of the received ZigBee is computed. Cross-technology precoding ensures immunity to signal distortion caused by hardware uncertainties in commodity devices. X-MIMO is evaluated on commodity devices of Atheros AR9334 and TelosB as well as on USRP B210 for in-depth analysis. Result demonstrates 495Kbps under <1% symbol error rate (SER) and 704.24 Kbps with 6.1% SER for two and three streams, where near-linear throughput improvement shows the effectiveness of X-MIMO.

To the best of our knowledge, X-MIMO is the first of its kind to offer MU-MIMO functionality on commodity IoT networks. In particular, X-MIMO effectively utilizes pervasively available WiFi infrastructure to bring IoT MU-MIMO into practical use, so as to immediately adoptable to billions of households and offices under zero cost. To summarize, our contribution is three-fold:

- We design X-MIMO, the first design to support MU-MIMO on commodity IoT devices without hardware or firmware modification. X-MIMO applies WiFi-ZigBee channel estimation and multiple precoded ZigBee signals emulation at commodity WiFi, which yields the different ZigBee packets at commodity ZigBee devices. X-MIMO is totally compatible with and easy to be deployed on commodity devices.
- To apply X-MIMO in practice, we address three practical challenges: precise timing control, hardware imperfection compensation, and multi-stream ZigBee signals emulation on 802.11n WiFi device. Moreover, our theoretical analysis shows that X-MIMO is immune to the phase uncertainty caused by the carrier frequency offset and hardware jitter.
- We implement and evaluate X-MIMO on commodity devices (TP-link WDR4300 wireless router with Atheros AR9334 and AR9580 WNIC and TelosB ZigBee mote) and USRP. Our experimental results demonstrate X-MIMO achieves reliable and high-throughput performances under line-of-sight and non-line-of-sight scenarios. In all the settings, the ZigBee symbol error rate is less than 1% and the throughput reaches above 495 Kbps, which is 2× of state-of-the-art WEBee [36]. In addition, our evaluation for three-stream X-MIMO reveals the near-linear throughput improvement showing the effectiveness of X-MIMO.

2 MOTIVATION

2.1 The Need for IoT MU-MIMO

This paper presents spectral efficient IoT down-link by enabling cross-technology multi-user MIMO. As the IoT reaches a massive scale and given the naturally limited resource of the wireless medium, it is critical to manage IoTs in a spectral efficient manner. We note that a large number of ZigBee/802.15.4 IoT devices are widely deployed to support variant applications across different sectors including smart homes and factories. Amazon Echo Plus, Samsung SmartThings, Philips Hue, Hive, Xiaomi Mijia, and IKEA Tradfri are among a large body of smart home gadgets. Smart factories often operate under 802.15.4-based protocols, such as WirelessHART [45], ISA100.11a [7], and TSCH [8]. For instance, Emerson's smart factory IoT network using WirelessHART is deployed at 54K smart factories worldwide, serving over 19 billion operating hours [5]. Managing massive scale IoT involves extensive traffic in controlling operation, updating the firmware for bug fixes, and reprogramming for failure recovery, The IoT traffic is anticipated to increase further with emerging applications such as AR/VR, where the intensive interactions incur heavy real-time traffic. Furthermore, the advancement in on-device AI is expected to increase the down-link traffic (as a tradeoff for reduced up-link) for downloading and updating the trained model. Achieving MU-MIMO for IoT at zero-cost only using the existing WiFi infrastructure is uniquely achieved by the two opportunities discussed in the following.

2.2 Opportunity #1: CTC

The emerging technique of CTC is a software-only solution enabling direct communication between commodity wireless running heterogeneous standards, without any hardware or firmware modification [34, 36, 52]. Communication between WiFi and ZigBee [36] is achieved by elaborately customizing the WiFi payload such that the transmitted WiFi signal is also interpreted and decoded as a ZigBee packet. This offers an opportunity to utilize the existing WiFi devices to manage IoT networks without introducing additional hardware cost. Furthermore, WiFi's high transmission power compared to low-power IoT offers an extended communication range advantageous in IoT management. However, the spectral efficiency of the state-of-the-art CTC is strictly constrained to single-input single-output (SISO), which essentially limits its capability in maintaining massive scale IoT. The current CTC designs simply waste the non-overlapped bandwidths between the transmitter and the receiver – 18 MHz between WiFi (20 MHz) and ZigBee (2 MHz) – further exacerbating the spectral efficiency. Taking CTC as a building block, X-MIMO builds MU-MIMO that fundamentally resolves such inefficiencies, paving a practical pathway to supporting massive IoT. This necessitates generating parallel CTC streams, which is enabled by the next opportunity.

2.3 Opportunity #2: Multi-antenna WiFi AP

In response to the higher throughput demand under the limited ISM spectrum (e.g., 100MHz on 2.4 GHz), WiFi has evolved to support various MIMO technologies. Widely deployed 802.11n WiFi APs are often equipped with multiple antennas (≥3) for MIMO functionality. From the signal processing standpoint, the antenna diversity is achieved by multi-stream data processing that enables simultaneous emission of separate waveforms from each antenna, in parallel. Multi-antenna and multi-stream processing are foundations to turning the WiFi AP into a MU-MIMO IoT transmitter in X-MIMO, where it essentially offers the opportunity for multi-stream CTC. We note that the number of antennas and streams are ever-increasing with the WiFi's evolution towards higher throughput. For instance, 802.11ax supports up to eight antennas and streams. This potentially offers extended opportunity and improved performance for X-MIMO.

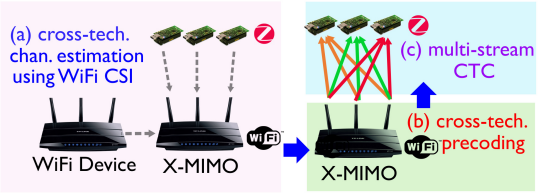


Figure 1: X-MIMO operates in three steps of (a), (b), and (c) to deliver multiple packets in parallel, up to the number of antennas and parallel streams supported by NIC.

3 X-MIMO OVERVIEW

Figure 1 illustrates the three steps of X-MIMO operation, to achieve MU-MIMO on commodity IoT using a WiFi AP: (a) In cross-tech. channel estimation, X-MIMO utilizes the WiFi fragmentation function to precisely control the timings of ZigBee and WiFi packets (from an arbitrary WiFi device associated with the X-MIMO WiFi AP) such that they overlap in time. This yields CSI measurement that reflects the overlapped ZigBee signal, from which X-MIMO recovers the received physical-layer ZigBee signal and further, the corresponding ZigBee channel. (b) In cross-tech. precoding, different ZigBee packets are precoded into multiple streams. such that upon the reception of the precoded streams, all the ZigBee devices are able to decode the different packets simultaneously. Then, (c) multi-stream CTC converts the precoded streams into a WiFi packet with the customized payload. Eventually, X-MIMO transmits this WiFi packet through multiple antennas on the commodity WiFi device and the ZigBee devices decode the different ZigBee packets simultaneously. Next, we introduce the MU-MIMO preliminaries for further understanding of the whole X-MIMO design. Lastly, we note that the number of ZigBee's that X-MIMO can support is throttled by the number of streams limited by the WiFi NIC hardware, which is typically 3 or higher (3 in our experimental device of TP-link WDR4300).

3.1 Preliminary: MU-MIMO

MU-MIMO supports multiple users to receive different signals simultaneously via precoding, which weighs each stream with an appropriate phase and amplitude according to the channel between AP and users. As a type of MU-MIMO, implicit MU-MIMO [39, 44, 48], uses up-link channel (Users to AP), which is estimated by AP, to perform precoding. In a typical implicit MU-MIMO scenario, as depicted in Figure 2, after the AP transmits the precoded signals $\boldsymbol{X} = [X_1, X_2]^\intercal$, it expects the two users to receive two independent streams, $\boldsymbol{S} = [S_1, S_2]^\intercal$:

$$\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \boldsymbol{H} \times \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \tag{1}$$

where $\boldsymbol{H}$ represents the estimated up-link channel. As a consequence, the AP obtains the precoded signals $\boldsymbol{X}$ via $\boldsymbol{X} = \boldsymbol{H}^{-1}\boldsymbol{S}$. Since it is impractical to extract channel information at commodity low-power IoT devices, X-MIMO adopts an implicit MU-MIMO approach, where precoding is performed at the X-MIMO side, incurring zero modification to the IoT devices, e.g., ZigBee.

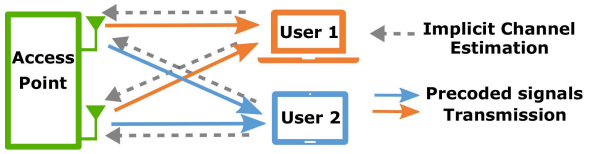


Figure 2: The AP estimates the up-link (Users to AP) channel, which is used to calculate the precoded signals, inducing the different packets at all users simultaneously.

4 X-MIMO DESIGN

To support implicit MU-MIMO, we introduce how the cross-tech. channel estimation is performed to implicitly collect the up-link channel information at X-MIMO, followed by the cross-technology precoding in this section.

4.1 Cross-technology Channel Estimation

Cross-technology channel estimation leverages the channel state information (CSI) provided by commodity WiFi WNICs, to obtain the ZigBee channel. CSI indicates the channel coefficient computed from the HT-LTF field of the WiFi preamble – i.e., by comparing the HT-LTF signal to what is received over the wireless channel:

$$CSI = \frac{Y}{X_w} \tag{2}$$

where X_w and Y are the WiFi HT-LTF and received signals in the frequency domain, respectively. We note that Y incorporates not only the HT-LTF but also other interfering wireless signals. Interestingly, physical-layer raw samples of such a signal can be recovered from CSI, by $Y = CSI \times X_w$. This serves as the fundamental idea behind cross-technology channel estimation – enforce ZigBee to interfere CSI, from which the ZigBee signal and channel are extracted. Next, we discuss this procedure in detail, followed by the mechanism carefully designed to be fully compatible with the WiFi and ZigBee standards and commodity devices.

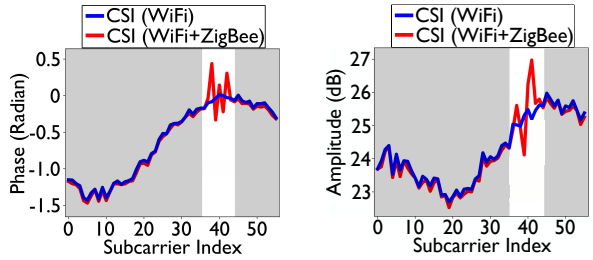
(a) WiFi packet (X_w) passes channel H_w. ZigBee packet (X_z) simultaneously arrives at X-MIMO via channel H_z.

(b) X-MIMO acquires CSI from the mixture of WiFi HT-LTF and the first 3.2 μs of ZigBee preamble.

Figure 3: WiFi CSI incorporates ZigBee channel upon signal overlap.

Figure 3(a) illustrates the scenario of obtaining the interfering ZigBee channel from the CSI measurement. H_w and H_z represent the WiFi and ZigBee channels, respectively, while X_z indicates the interfering ZigBee signal. Under this scenario the signal at X-MIMO becomes the mixture of the WiFi and ZigBee signals received through the corresponding channels, yielding $Y = H_w X_w + H_z X_z$. Plugging this into Eq. 2 we get

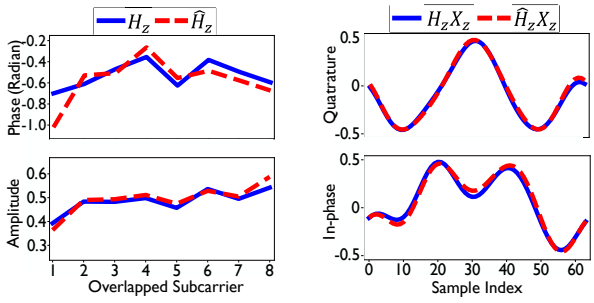
$$H_z = \frac{X_w(CSI - H_w)}{X_z} \tag{3}$$



(a) Phase. (b) Amplitude.

Figure 4: CSI of the WiFi + ZigBee packet vs. WiFi channel H_w. ZigBee overlaps with the WiFi subcarriers in white.

Doing so indicates that H_z can be computed as the RHS is entirely known: (i) X_w is the standard WiFi HT-LTF, a known signal. (ii) H_w can be found from the previously received WiFi packet, within the coherence time. Lastly, (iii) X_z is also a known signal under the accurate timing in Figure 3(b). That is, by aligning the beginning of the ZigBee packet with WiFi HT-LTF, X_z becomes the first 3.2 μs (i.e., WiFi HT-LTF duration) of the ZigBee preamble, which is known. This effectively demonstrates that cross-technology (i.e., ZigBee) channel can be estimated using WiFi CSI.



(a) Computed ZigBee Channel. (b) Computed ZigBee Signal.

Figure 5: Received ZigBee signal ($\widehat{H}_zX_z$) and estimated channel ($\widehat{H}_z$) computed from CSI, compared with ground truths (H_zX_z and H_z).

To further understand this intuition, Figure 4 depicts an experimental example of *CSI* phase and amplitude, compared to H_w. The figure shows that subcarriers overlapping with the ZigBee vary significantly, as the *CSI* incorporates ZigBee (X_z) signal and channel (H_z). Meanwhile, the non-overlapped subcarriers (in gray) remain consistent. We note that the Figure 4 is obtained after compensating the offsets that inevitably occurs in practical systems. For brevity, we rigorously discuss the compensation algorithm in Appendix. Figure 5(a) illustrates the computed ZigBee channel, denoted by $\widehat{H}_z$ and Figure 5(b) demonstrates the computed ZigBee signal (i.e., $\widehat{H}_zX_z$) . They closely approximate the ground truth, providing empirical validation of our technique.

Until now we have demonstrated cross-technology channel estimation under the condition that the first 3.2 μs ZigBee preamble (i.e., X_z) precisely overlaps with the WiFi HT-LTF. In practice, the requirement of such strict timing control is inherently difficult to satisfy. This is because the commodity devices running contention-based MAC protocols (i.e., CSMA), including WiFi and ZigBee, have uncontrollable channel access delays. In the following, we discuss X-MIMO's unique and highly precise timing control mechanism under practical settings.

4.2 Timing Control via WiFi Fragmentation

X-MIMO's timing control only uses standard-defined functionalities for full compatibility to commodity WiFi and ZigBee – therefore it is, (i) non-disruptive to coexisting networks, (ii) does not require any modification to the firmware or driver, and (iii) is very light-weight, as it does not involve any extra coordination or time synchronization protocols. Further, the timing control operates under a typical WiFi network setting where a WiFi device is associated to a WiFi AP (running X-MIMO). This indicates a wide applicability.

Timing WiFi and ZigBee signals to precisely overlap in time leverages the WiFi packet fragmentation function. Commonly provided in WiFi NICs, this function cuts down a large fragment to smaller pieces where the fragment interval is precisely kept at 60 μs (=2×SIFS (16 μs) + WiFi ACK duration). Meanwhile, ZigBee ACK is triggered exactly 192 μs (macSifsPeriod) after a packet reception. They are both strictly enforced by the standards [3, 6] on commodity devices and serve as our basis to precise time control. We note that WiFi packet fragmentation can be simply set using `iwconfig` (under Linux) command, without involving any hardware, firmware, or driver modifications.

(a) WiFi Fragment 1 (embedding ZigBee CTC packet) is received by both ZigBee and X-MIMO (who retrieves H_w).

(b) WiFi Fragment 2 and ZigBee ACK overlap from which X-MIMO retrieves H_z.

Figure 6: Timing control via fragmentation.

Figure 6 illustrates how the timing works using two fragmented WiFi packets. As in Figure 6(a), the WiFi device transmits the first fragmented WiFi packet, which emulates a ZigBee packet[1] (i.e., CTC). Note that this is a legitimate WiFi packet encapsulating a ZigBee packet in its payload – therefore, it is received by both WiFi (i.e., X-MIMO) and the ZigBee. Upon receiving this packet, X-MIMO obtains the WiFi channel (WiFi device→X-MIMO) estimation, H_w. Meanwhile, packet reception at ZigBee triggers an ACK, as defined in the standard. We note that the entire process leverages the standard MAC protocol and thus does not require coordination between WiFi and ZigBee.

As shown in Figure 6(b), the second fragment from the WiFi device and the ZigBee ACK simultaneously arrive at X-MIMO. This is because both the WiFi fragment and the ZigBee ACK are transmitted with a fixed delay. Furthermore, this overlapped packet is highly likely to be correctly received at X-MIMO given WiFi's significantly higher power. The ZigBee channel, $\widehat{H}_z$, is computed using the CSI

[1]The target ZigBee device ID can be easily obtained by the existing ZigBee network.

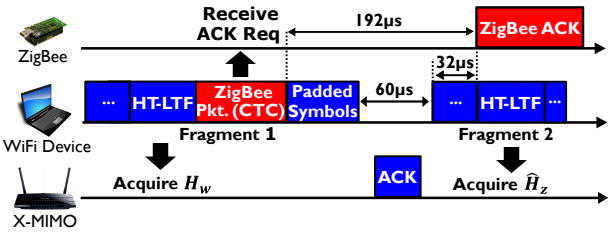


Figure 7: Detailed timing in ZigBee channel estimation. ZigBee ACK is aligned to HT-LTF (Fragment 2) by controlling the padded symbol duration to 100 μs, i.e., ZigBee ACK delay (192 μs) = padded symbol duration (100 μs) + fragment interval (60 μs) + preamble duration (32 μs, before HT-LTF).

from this packet. We note that the CSI and the estimation results demonstrated in Figures 4 and 5 are obtained from commodity WiFi and ZigBee, using this timing mechanism.

Detailed timing. Figure 7 presents the timing details in the channel estimation process. The beginning of the ZigBee ACK is aligned to the HT-LTF of the second fragment, so as to exploit the known ZigBee preamble for ZigBee channel estimation. To do so, symbols are padded after the ZigBee CTC packet such that padded symbol duration, fragment interval (60 μs), and preamble duration (32 μs) add up to ZigBee ACK delay (192 μs). This results in the 100 μs-long padded symbols. Since a WiFi symbol is 4 μs long (including CP), a total of 25 symbols are padded. In practice, this is done by simply setting the fragmentation threshold to the size of the ZigBee CTC packet plus the padded symbols. The overhead for the channel estimation remains light, as (i) the padded symbols can be used for data delivery from the WiFi device to X-MIMO and need not be wasted, and (ii) the narrow-band 2.4 GHz ZigBee channel remains consistent (i.e., the coherence time) over seconds [40, 49], far longer than wider-band systems such as WiFi ($\sim 40ms$) coherence time [56]. Given that ZigBee packets are typically $\sim$ 1 ms long, thousands of ZigBee packets can be delivered following an estimation. Channels of multiple ZigBees may be obtained separately by simply repeating the process. Alternatively, the channels may be rapidly estimated back-to-back by increasing fragments, for instance to 4, 6, or more, where two fragments are consumed per channel estimation.

4.3 Cross-technology Precoding

Here we discuss how uplink channel estimation is applied to precoding to achieve MU-MIMO. As in Figure 8, for simplicity we present our design for two ZigBee receivers, where it can be straightforwardly extended to support more users as evaluated in Section 6.3. Let us denote the two ZigBee packets to be delivered as $Z = [Z_1, Z_2]^\intercal$. By directly applying the estimated uplink channel $\widehat{\boldsymbol{H}}_{\boldsymbol{z}} = \begin{pmatrix} \widehat{h}_{11} & \widehat{h}_{12} \\ \widehat{h}_{21} & \widehat{h}_{22} \end{pmatrix}$ into Eq. 1, the precoded signal is computed as:

$$X = \widehat{\boldsymbol{H}}_{\boldsymbol{z}}{}^{-1} Z \tag{4}$$

Obviously, the quality (i.e., SNR) of the signals received at the ZigBee devices reflects the precision of channel estimation. In other words, X-MIMO's performance is largely affected by the accuracy of $\widehat{\boldsymbol{H}}_{\boldsymbol{z}}$. As X-MIMO operates on commodity devices only using standard functions, $\widehat{\boldsymbol{H}}_{\boldsymbol{z}}$ incorporates inevitable phase errors that are unique in X-MIMO. The error stems from three sources: (i)

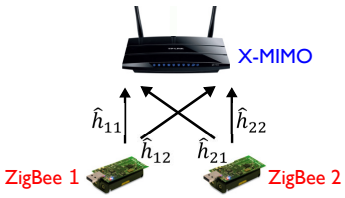


Figure 8: X-MIMO directly leverages the estimated up-link channels to achieve MU-MIMO. The two ZigBee scenario is shown for simplicity where X-MIMO supports as many as the number of antennas on the WiFi AP.

carrier frequency offset (CFO) between the ZigBee and X-MIMO. This is because X-MIMO locks its carrier frequency to the WiFi device (via phase-locked loop), not ZigBee. Therefore, CFO between ZigBee and WiFi persists. (ii) The initial phase offset between X-MIMO and ZigBee. Lastly, (iii) the jitter in the ZigBee ACK arrival time (< 0.1 μs in our experiment) introduces an additional phase error. While these uncertainties may add up to a large phase error, interestingly, it has zero impact on the ZigBee signal quality. That is, X-MIMO is inherently immune to such uncertainties, which we prove via a rigorous derivation.

We refer back to the example scenario in Figure 8 involving two ZigBees, where we first consider the case for ZigBee 1. Let us denote the jittered ACK (from ZigBee 1) reception time and the timing jitter from that time as t_1 and τ_1, respectively. We further indicate the CFO and the phase offset between X-MIMO and ZigBee 1 as Δf_1 and θ_1, respectively. Letting $\angle p_1^k$ represent the total phase change incurred on ZigBee 1 channel estimation on the subcarrier k, this becomes:

$$\angle p_1^k = \underbrace{2\pi\Delta f_1 t_1}_{\text{CFO}} + \underbrace{\theta_1}_{\text{offset}} + \underbrace{2\pi k f_\delta \tau_1}_{\text{ACK jitter}} \tag{5}$$

where $1 \le k \le 64$ (subcarrier index). We note that p_1^k, a complex value with the amplitude of 1, is embedded in the corresponding subcarriers of $\widehat{h}_{11}$ and $\widehat{h}_{12}$ (in Figure 8), as they are obtained from the same ACK sent by ZigBee 1. Similarly, the phase uncertainty for ZigBee 2 for subcarrier k is $\angle p_2^k$, where this is included in $\widehat{h}_{21}$ and $\widehat{h}_{22}$. By denoting phase shifts for ZigBee 1 and 2 in all subcarriers as $\angle p_1$ and $\angle p_2$, the relationship between the estimated and the true channel can be represented as $\widehat{h}_{11} = p_1 h_{11}$ and $\widehat{h}_{12} = p_1 h_{12}$ (similar for ZigBee 2). Therefore,

$$\begin{aligned} \widehat{\boldsymbol{H}}_{\boldsymbol{z}} &= \begin{pmatrix} p_1 h_{11} & p_1 h_{12} \\ p_2 h_{21} & p_2 h_{22} \end{pmatrix} \\ &= \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \\ &= \boldsymbol{P} \times \boldsymbol{H}_{\boldsymbol{z}} \end{aligned} \tag{6}$$

where $\boldsymbol{H}_{\boldsymbol{z}}$ is the ground truth ZigBee channel and $\boldsymbol{P} = \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix}$.

X-MIMO's precoded signal, $\boldsymbol{X}$, is designed to yield $\boldsymbol{Z}$ after passing through the estimated channel, $\widehat{\boldsymbol{H}}_{\boldsymbol{z}}$ – i.e., $\widehat{\boldsymbol{H}}_{\boldsymbol{z}}\boldsymbol{X} = \boldsymbol{Z}$. Applying Eq. 6 and solving for $\boldsymbol{X}$ we get

$$\begin{aligned} \boldsymbol{X} &= (\boldsymbol{P}\boldsymbol{H}_{\boldsymbol{z}})^{-1} Z \\ &= \boldsymbol{H}_{\boldsymbol{z}}^{-1} \boldsymbol{P}^{-1} Z \end{aligned} \tag{7}$$

In reality, $\boldsymbol{X}$ passes through the channel $\boldsymbol{H_z}$ to reach ZigBee 1 and 2. Therefore, what is received by the ZigBee devices when X is the transmitted (precoded) signal, is:

$$\begin{aligned} \boldsymbol{H_z X} &= \boldsymbol{H_z H_z^{-1} P^{-1} Z} \\ &= \boldsymbol{P^{-1} Z} \\ &= \begin{pmatrix} p_1^{-1} & 0 \\ 0 & p_2^{-1} \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \\ &= \begin{pmatrix} p_1^{-1} Z_1 \\ p_2^{-1} Z_2 \end{pmatrix} \end{aligned} \tag{8}$$

where we used Eq. 7. This indicates that received signals at the two ZigBee 1 and 2 are simply phase-shifted (by $\angle p_1^{-1}$ and $\angle p_2^{-1}$) versions of the target signal $\boldsymbol{Z}$. The phase-shifted signal does not impact on ZigBee reception – this is because the commodity ZigBee receivers only rely on the phase *differences* (i.e., the relative phase) between symbols for decoding [21]. Thus, the rotation of the entire signal is harmless. This effectively demonstrates that X-MIMO is indeed immune to the unique and inevitable uncertainties in the cross-technology channel estimation. The evaluation in sections 6.2 and 6.5 empirically validate the immunity to phase shifts and demonstrates high robustness of the cross-technology channel estimation under various channel conditions including LOS and non-LOS scenarios.

Power Control. X-MIMO is based on the principle of the implicit MU-MIMO, which requires the transmission power of the AP to be identical to that of the users. Correspondingly, X-MIMO sets the transmission power of the precoded ZigBee (8 subcarriers in X-MIMO) and ZigBee devices to the same level, using the `iwconfig` command (in Linux), without driver or firmware modification. Specifically, X-MIMO sets the transmission power of each antenna to be 8.45 dB higher than the transmission power of the ZigBee device. This is because only eight WiFi subcarriers (among a total of 56) corresponds to the ZigBee, occupying 8/56 of the WiFi signal. In other words, the power of the entire WiFi signal (emitted from each antenna) should be higher than the target ZigBee signal power by 8.45 dB (=$10log_{10}(8/56)$). Therefore WiFi AP equipped with two antennas should set up to overall power of 11.45 dB (twice of 8.45 dB). For instance, the default transmission power of CC2530 radio chip (ZigBee) is 4.5 dBm [1]. Then, the transmission power of each antenna at X-MIMO should be 12.95 dBm for the total transmission power of 15.95 dBm for two antennas. As 15.95 dBm is close to the default transmission power of typical WiFi at $\sim$ 17 dBm (e.g., AR9334), the impact of X-MIMO's power control to WiFi communication is insignificant.

5 MULTI-STREAM CTC

Multi-stream CTC is uniquely designed to transmit the precoded signal on a commodity WiFi AP (X-MIMO), by leveraging its 802.11n MIMO features and functionality – i.e., multiple antennas and multi-stream signal processing. Compared with the latest CTC designs, X-MIMO significantly improves the flexibility of signal manipulation from a single stream to multiple streams. Furthermore, X-MIMO incorporates spectral efficient emulation to avoid the spectral wastage in the state-of-the-art CTC, caused by the unused subcarriers.

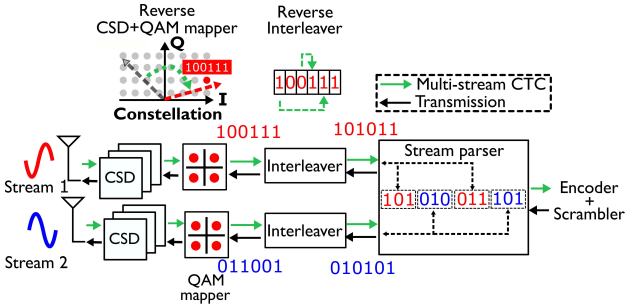
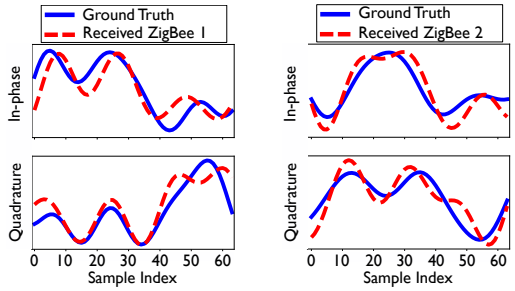


Figure 9: The encoded bits are obtained to emulate the multi-stream signals via reversing CSD, QAM mapper, Interleaver, and Stream parser.

Multi-stream CTC is achieved by reversing each step of 802.11n transmission to find an appropriate 802.11n payload such that the corresponding transmitted signal is the precoded signal, or equivalently using an 802.11n signal to emulate the precoded signal. As illustrated in Figure 9, our first step is to reverse the Cyclic Shift Diversity (CSD), which is inserted to prevent unintended beamforming via multiplying the QAM mapped signal on the subcarrier k, stream i by a complex value c_i^k (= $e^{j\pi \frac{k \bmod 8(i-1)}{4(i-1)}}$, for i $\geq$ 2, 1 for all other i) [2]. Specifically, reversing CSD is simply performed by multiplying the precoded signal on the subcarrier k stream i by the conjugate of c_i^k to obtain the corresponding QAM mapped signal, denoted by q_i^k. To approximate q_i^k with minimum error, the closest QAM sample in the constellation diagram is selected, yielding a bit sequence for each signal stream. As the 'Reverse CSD+QAM mapper' in the Figure 9 illustrates, the red arrow is computed from the gray arrow (precoded signal) by reversing the CSD. The red arrow is then approximated by the red QAM sample, from which the bit sequence '100111' is generated.

Since the interleaver shuffles the bit sequence deterministically, reversing the interleaver is performed via rearranging each bit sequence accordingly. The stream parser, cutting the serial encoded bits into multiple blocks and further feeding to multiple bit sequences, is reversed by assembling bits into the blocks and placed in the encoded bits alternatively. For instance, as 'Reverse Interleaver' in the Figure 9 illustrates, indices 3,4 and 1,5 in the red bit sequence '100111' are switched to yield '101011'. Then, every three bits in the red sequence are assembled into one block and further placed at the odd index while the blocks from the blue sequence are placed at the even index in the encoded bits. Consequently, the serial encoded bits '101010011101' are generated from the two streams.

To finally obtain the payload for multi-stream CTC, we need to reverse the encoding and scrambling, which are the first two steps in 802.11n transmission. Since the principle of encoder and scrambler are equivalent for 802.11g and 802.11n, we adopt the design in WEBee [36] (designed for 802.11g) to convert the encoded bits to the payload, except that the scrambler seed, a 7-bits sequence controlling the scrambler, cannot be manually set in 802.11n. To resolve this issue, we take advantage of the predictable seed sequences (increments by one between two packets) in many commodity WiFi chips (e.g., AR9334 and AR9380) and the fixed initial scrambling seed (seed index 71 out of 128). Consequently, given the count of the



(a) Ground truth and received signal at ZigBee device 1.

(b) Ground truth and received signal at ZigBee device 2.

Figure 10: Comparison between received signals and ground truth at two ZigBee devices.

transmitted WiFi packet, the scrambling seed in 802.11n is easily tracked [2]. Upon finding the scrambling seed, the scrambler can be reversed to yield the payload for multi-stream CTC.

For a further intuition on the entire design, Figure 10 demonstrates the ZigBee signals at the two receivers in comparison to the ground truth. The slight difference indicates our design inherits the limitation of the state-of-the-art CTC – determined by the finite constellation points, the precision is degraded by emulation errors. Despite the inevitable error, the successful operation of X-MIMO is promised by the high redundancy in ZigBee's direct sequence spread spectrum (DSSS).

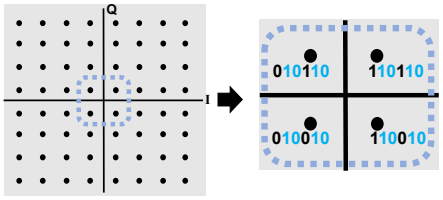


Figure 11: The bits, associated with the weakest samples in QAM 64 constellation diagram, reveal a pattern: the bits at indices 2–3 and 5–6 are '1010'.

5.1 Spectral Efficient Emulation

Due to the asymmetric bandwidth (20 MHz in WiFi and 2 MHz in ZigBee), the large spectral wastage (18 MHz) degrades the spectral efficiency. To resolve this issue, we select the weakest QAM samples in the constellation diagram (in "Reverse CSD+QAM mapper) to suppress the power allocated to the subcarriers (non-overlapped with ZigBee), thereby opening the unused frequencies for other wireless devices. As depicted in Figure 11, we discover that if the six bits allocated to this subcarrier are 'X10X10', where 'X' is an arbitrary '0' or '1', the generated constellation QAM sample (using QAM 64) are weakest. Therefore, we enforce the six bits on each non-overlapped subcarrier to follow the form of 'X10X10' in our Multi-stream CTC to minimize the energy leakage to the unused frequencies.

Figure 12 shows an example of the power spectral density (denoted by Power Spec.) associated with the waterfall of the 802.11n packets, which emulate precoded signals on two ZigBee channels

[2]We experimentally validated scrambling seed tracking on various commodity WiFi devices running different firmware versions under Ubuntu 12.04 and 14.04. E.g., TP-Link TL-WDR4300 router (AR9334, AR9580), COMPEX WLE350NX (AR9580) and Atheros AR5BXB112 (AR9380) WiFi NICs

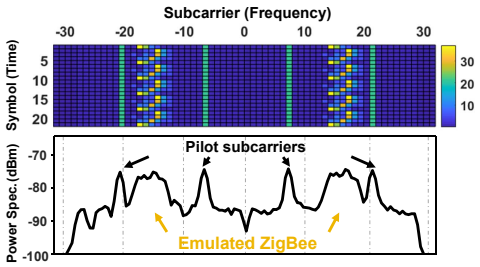


Figure 12: Waterfall and power spectral density.

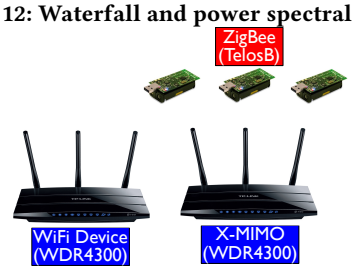


Figure 13: Our system consists of two WiFi wireless routers (one works as X-MIMO and the other transmits two customized WiFi fragments), and telosB nodes (ZigBee).

(±5 MHz apart from the center frequency). As demonstrated in Figure 12, we discover that the non-overlapped subcarriers' power (suppressed by emulation) is 15.68 dB less than the peak subcarriers' power, validating the effectiveness of our spectral efficient emulation. Although the pilot subcarriers (represented by the high power of four lines and peaks in Figures 12) are uncontrollable (thus, not suppressed), their impacts are negligible (demonstrated in Section 6.4).

6 EVALUATION

This section discusses implementation of X-MIMO on commodity devices and the performance analysis under practical scenarios.

6.1 Implementation

Figure 13 illustrates our implementation setup with two TP-link TL-WDR4300 wireless routers and three Telosb nodes. A router is running X-MIMO while the other operates as a common WiFi device, and the TelosB nodes are ZigBee nodes. The details of the X-MIMO implementation[3] are as follows: X-MIMO modifies the ath9k-based Atheros CSI tool [56] to support the large maximum transmission unit (MTU) for emulating long ZigBee streams. We set another TP-link TL-WDR4300 wireless router to inject customized WiFi fragmented packets. To implement multi-stream emulation on top of 802.11n physical layer, we track the transmission of each WiFi packet ever since the WiFi is initialized on the TP-link WDR4300 router in order to track the scrambler seed. In our experiments, we upload these customized packets to the wireless router and inject them via `lorcon`. Since tracking the scrambler seed is just one step from system initialization while the TP-link WDR4300 router has a 560 MHz CPU and 128 MB RAM, the overhead is negligible.

To evaluate the performance of X-MIMO, we modify TinyOS to access the raw received bits at two ZigBee nodes. We disable the CRC check via setting the register "MODEMCTRL0.AUTOCRC" [6],

[3]X-MIMO codes to be released at https://github.com/smilelabkaist/x-mimo.

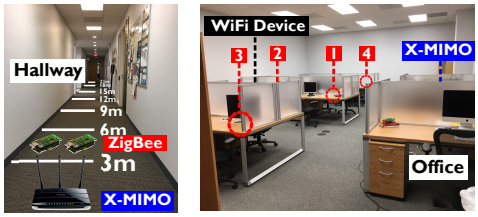


Figure 14: X-MIMO is evaluated at two places: hallway (line-of-sight) and office (non-line-of-signt).

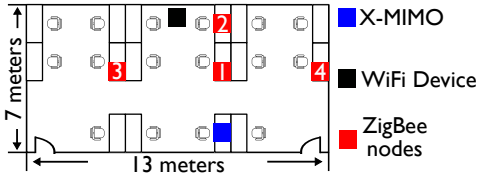


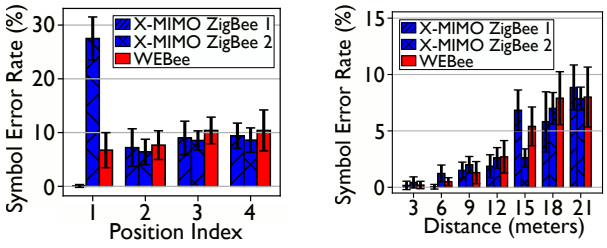
Figure 15: The specific deployment of X-MIMO, the WiFi device (transmits customized fragments) and ZigBee devices in the office.

which is commonly supported on all TI serial radio chips and use `Printf` interface (in TinyOs library) to print all the received raw symbols. We note that disabling CRC is only for analysis purposes; i.e., obtaining the symbol error rate and calculating the corresponding throughput. In addition to the above implementations, we use USRPs as ZigBee nodes to test our ZigBee channel estimation and the performance of X-MIMO under different settings. We modify the 802.15.4 implementation [4] on GNURadio to further check the detailed ZigBee symbol error rate at USRP.

6.2 X-MIMO Performance

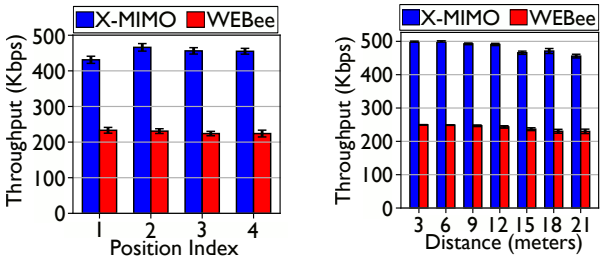
In this experiment, we evaluate the performance of X-MIMO under two TelosB nodes (commodity ZigBee) while X-MIMO with more ZigBee receivers is evaluated in Section 6.3. Figure 14 depicts the two scenarios in our evaluation: hallway and office. To evaluate the performance of X-MIMO in the line-of-sight (LOS) scenario, we deploy the X-MIMO device and ZigBee devices at different distances in the hallway. To evaluate X-MIMO in the non-line-of-sight (NLOS) scenario, we deploy ZigBee devices at four different positions in the office, which is shown in Figure 15. In this experiment, the transmission power of the ZigBee device is 0 dBm and the transmission power of the WiFi fragments is the default 17 dBm. We set the Tx power of X-MIMO according to our design in Section 4.3. The symbol error rate and throughput of X-MIMO are evaluated and compared with WEBee [36].

Symbol Error Rate (SER). The ZigBee symbol error rate of X-MIMO and WEBee in LOS and NLOS scenarios are shown in Figure 16. In our experiment, WEBee transmits to the two ZigBee devices alternatively and therefore its SER is the average of the SER at two ZigBee devices. The SER of X-MIMO for two ZigBee devices at position 1 is 1% and 27%, exhibiting a significant imbalance. This is because the channel from X-MIMO to ZigBee 2 is so weak that the signal for ZigBee 1 keeps dominating the ZigBee device 2. The SER of X-MIMO at the other three positions are (9.1%, 7.2%), (9%, 8.5%) and (9.4%, 8.6%) while the SER of WEBee is 7%, 10.1% and 10.2, respectively.



(a) Non-line-of-sight Scenario. (b) Line-of-sight Scenario.

Figure 16: Symbol error rate of X-MIMO and WEBee in NLOS and LOS scenario.



(a) Non-line-of-sight Scenario. (b) Line-of-sight Scenario.

Figure 17: Throughput of X-MIMO and WEBee in NLOS and LOS scenario.

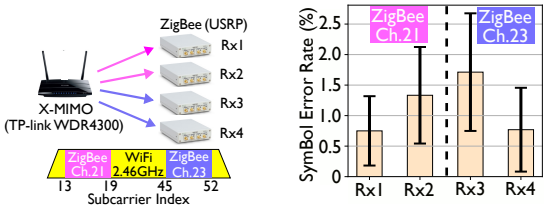
In the LOS scenario, two ZigBee devices are placed at distances of 3 - 21 m away from the X-MIMO. For the distance of 12 and 18 m, SER of X-MIMO for two ZigBee devices is less than WEBee. Despite the small imbalance of SER at two ZigBee devices (6.8% at ZigBee 1 and 2.6% at ZigBee 2), the average SER of X-MIMO at 15 m distance is 4.7%, which is still less than the SER of WEBee (5.4%). As the distance between X-MIMO and ZigBee devices increases, the SNR of the overlapped ZigBee signal gets weaker while the SER of X-MIMO does not drop too much. X-MIMO still achieves $\leq 9\%$ SER with $\leq 2\%$ error at 21 m.

Throughput. Figure 17 demonstrates the throughput of X-MIMO obtained from the SER. In the NLoS scenario, X-MIMO achieves 432, 466, 456, and 455 Kbps at four positions, outperforming WEBee by $\times 1.85 - 2.03$. In the LOS scenario, the throughput of X-MIMO exhibits a more stable trend. At the distance of $\leq$ 12 meters, the throughput of X-MIMO is greater than 490 Kbps, which is almost doubling the throughput of legacy ZigBee. For the distance of 15, 18, and 21 meters, the throughput of X-MIMO (465, 471, and 455 Kbps) are almost twice of WEBee. This result shows that given the two-stream precoding in X-MIMO, the throughput of communication for IoT is significantly improved by the number of antennas.

6.3 Scalability of X-MIMO

To demonstrate the scalability of X-MIMO, we extend the implementation of two-streams X-MIMO to support two parallel ZigBee channels and three streams (three ZigBee receivers).

Parallel X-MIMO. The settings of parallel X-MIMO are illustrated in Figure 18(a): (i) X-MIMO is implemented on TP-link WDR4300 wireless router, which works on 2.46 GHz to cover ZigBee channel 21 (2.455 GHz) and 23 (2.465 GHz). Despite 2.46 GHz is not the center frequency of any WiFi channel, the ath9k WiFi driver and the commodity WiFi device allow us to set the center frequency

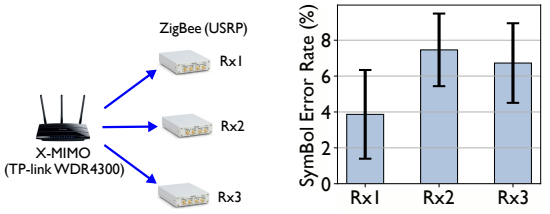


(a) Experiment setting for parallel X-MIMO.

(b) Symbol error rate of four ZigBee devices on two channels.

Figure 18: Performance of parallel X-MIMO.

to be an arbitrary value (1 MHz granularity) via controlling register 'channelSel' in the function 'ar9003_hw_set_channel'. We control the emulation process to transmit the precoded signals on subcarriers 13 - 19 and 45 - 52 to support two-stream MU-MIMO on ZigBee channels 21 and 23. (ii) Two ZigBee devices (USRPs) are deployed on each ZigBee channel. The distance between X-MIMO and the four ZigBee devices (Rx1 - Rx4) is 3 meters. The symbol error rate of parallel X-MIMO is demonstrated in Figure 18(b), showing that the SER of four devices is less than 1.7%. Hence, with the two-channel parallel X-MIMO, we enable MU-MIMO for four ZigBee devices with an aggregated throughput of 983 Kbps.



(a) Experiment setting for three-streams X-MIMO.

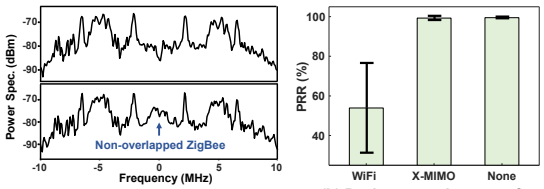
(b) Symbol error rate of three ZigBee devices .

Figure 19: Performance of three-streams X-MIMO.

Three-stream X-MIMO. In this experiment, the performance of three-streams X-MIMO is evaluated on the TP-link 4300 wireless router with three antennas. Specifically, the TP-link WDR4300 router is equipped with two WNICs, i.e. AR9344 (2.4 GHz) and AR9580 (5 GHz). The AR9344 WNIC only supports up to two antennas while the AR9580 WNIC supports three antennas. Then, we implement X-MIMO on AR9580 (5 GHz) as a workaround to demonstrate the performance of three-streams X-MIMO. Specifically, as Figure 19(a) illustrates, we set X-MIMO (AR9580 WNIC on TP-link WDR4300) to work on WiFi channel 44 (i.e., 5.22 GHz) and deploy three USRPs, running ZigBee module, on 5.225 GHz as MU-MIMO receivers. The distance between X-MIMO and three ZigBee receivers (Rx1 - Rx3) is 2 meters. The performance of three-streams X-MIMO is shown in Figure 19(b). The symbol error rate at three ZigBee receivers is less than 7%, while the average SER is 6.1%.

6.4 X-MIMO Spectral Efficiency

We evaluate our proposed spectral efficient emulation to show the improvement in the spectral efficiency. Specifically, two ZigBee devices (Tx and Rx), working on X-MIMO's non-overlapping frequencies, are deployed 4.5 meters apart while a USRP (X-MIMO) is placed 2.35 meters away from ZigBee Rx. The Tx power of ZigBee and USRP is set to be 5 dBm. The spectral efficient emulation in practice is shown in Figure 20(a), where the leakage to the non-overlapped frequencies is -80 dBm. Such leakage is weaker than



(a) The Power Spectrum of X-MIMO while ZigBee is transmitted on the non-overlapped frequencies.

(b) Packet reception rate of ZigBee under three different scenarios.

Figure 20: ZigBee communication on the unused frequencies under different scenarios.

the default CCA threshold on the typical commodity ZigBee device such as CC2420 (i.e., -71 dBm)[6], thereby incurring zero impact on the ZigBee Rx's communication. The effectiveness of spectral reuse is shown in Figure 20(b) through the packet reception ratio under three wireless scenarios: (i) The USRP transmits regular WiFi packets, which overlap with ZigBee channel, (ii) The USRP (X-MIMO) transmits WiFi packets with spectral efficient emulation in the same frequency band, and (iii) The USRP does not transmit any wireless signals. While ZigBee Rx receives 53.92% packets due to severe interference under (i) WiFi scenario, where ZigBee Rx shows 99.28% and 99.50% of packet reception rate, under (ii) X-MIMO and (iii) none scenarios, respectively. This experiment shows that X-MIMO with spectral efficient emulation is able to avoid the spectral wastage on the non-overlapped subcarriers. Thus, the maximum spectral efficiency X-MIMO can achieve is to 0.36 bits/s/Hz, 3× of the legacy ZigBee, and 28.8× of the state-of-the-art design WEBee[36].

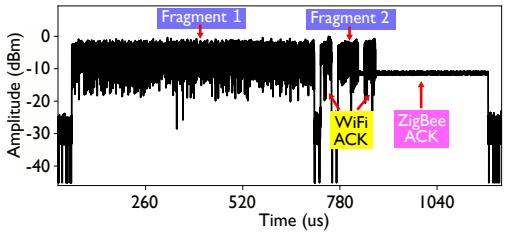
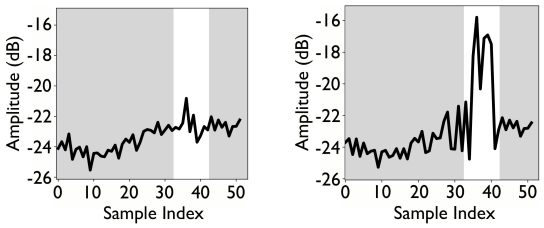


Figure 21: Amplitude of the signal generated in the cross-technology channel estimation.

6.5 Cross-tech. Channel Estimation in Practice

In this Section, we measure and show the real traffic associated with and the performance of cross-technology channel estimation.

Timing Control in Practice. We deploy one USRP to measure the signal generated at X-MIMO, ZigBee device, and the WiFi device. We set the fragmentation threshold to be 1898 Bytes (= 146 WiFi MCS 3 symbols), which consists of 1 symbol for MAC header, 120 symbols for emulating ZigBee ACK request, and 25 padded symbols. Then, the duration of Fragment 1 is 36 (preamble) + 146×4 (146 symbols in the payload) = 620 μs. Figure 21 depicts the traffic of cross-technology channel estimation consists of two WiFi fragments, the replied WiFi and ZigBee ACKs. After the WiFi device transmits the Fragment 1 (620 μs), the X-MIMO responses with a WiFi ACK, followed by Fragment 2, which collides with the replied ZigBee ACK (359 μs). Given that the inter-fragment interval is 60 μs, the whole time consumed for cross-technology channel estimation is 620 + 60 + 32 + 359 = 1071 μs, which is is negligible compared to the long coherence time for narrow-band ZigBee signal (over seconds[40, 49]).



(a) Amplitude of CSI collected from ZigBee devices placed 21m away.

(b) Amplitude of CSI collected from ZigBee devices placed 6m away.

Figure 22: The amplitude of CSI with ZigBee overlapping in the LoS scenario.

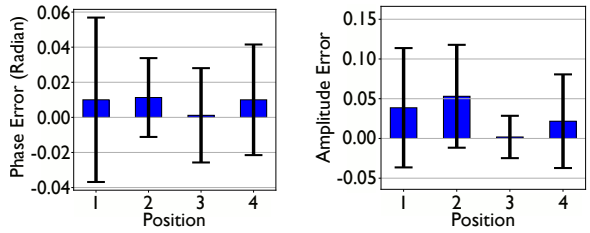
Cross-tech. Channel Estimation Precision. We use two USRPs in this experiment, where USRP 1 works as X-MIMO, and USRP 2 transmits customized WiFi fragments on antenna 1 and works as a ZigBee device on antenna 2. The HT-LTF field in the transmitted WiFi fragments and the ZigBee signal are perfectly aligned. To get the ground truth of ZigBee channels, we also control the USRP 2 to transmit ZigBee signal without the interference of WiFi fragments to USRP 1, where the ground truth ZigBee channel is obtained by comparing the first 3.2 μs received ZigBee signal and the first 3.2 μs transmitted ZigBee signal. The Tx power of WiFi fragments (USRP) is set to be 17 dBm and the Tx power of ZigBee (USRP) is set to be 0 dBm. We deploy the two USRPs in the same LOS and NLOS scenario as Figure 14 illustrates and compare the estimated ZigBee channel with ground truth.

In our experiment, the position of the WiFi device (transmits customized fragments) does not change with the ZigBee devices, thus leading to the constant amplitude of the WiFi channel. However, the ZigBee signal strength varies with its position. For example, Figure 22 shows two CSI values (with ZigBee overlapped) collected from the ZigBee 6 and 21 meters away from X-MIMO. The ZigBee signal at 21 meters is much weaker than the ZigBee signal at 6 meters.

Method. The absolute phase of the estimated ZigBee channel is affected by the hardware uncertainty, resulting in a time-variant estimation compared to the ground truth. Hence, in this experiment, we use the relative phase between two ZigBee channels as the metric to check the precision of the phase of the estimated ZigBee channel. Since the phase of two estimated ZigBee channels is affected simultaneously by the hardware uncertainty as described in Section 4.3, Eq. 6 indicates that the relative phase between two estimated ZigBee channels are immune to the hardware uncertainty. Specifically, the relative phase between the estimated channel p_1h_{11} and p_1h_{12} is identical with $\angle(h_{11}, h_{12})$. Thus, the relative phase is kept the same within channel coherent time.

To evaluate the precision of the amplitude of the estimated ZigBee channel, we utilize the amplitude ratio between two estimated channels as the metric. Specifically, the amplitude ratio between the estimated channel p_1h_{11} and p_1h_{12} is $|h_{11}|/|h_{12}|$, which removes the influence of hardware uncertainty p_1. Hence, the amplitude ratio is an indicator to check the precision of the estimated ZigBee channel, in terms of amplitude.

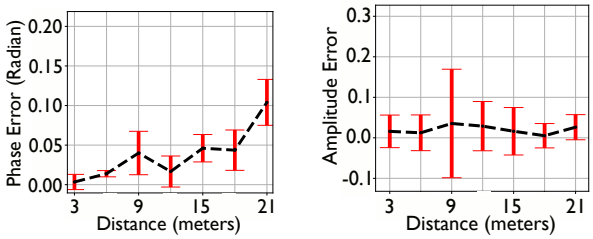
Results. Figure 23(a) illustrates the precision of the relative phase between two estimated ZigBee channels at four positions in the office. The error in the relative phase of the estimated ZigBee channel is ≤ 0.013 rad, with the maximum standard variance of 0.04 rad, indicating that the phase of the estimated ZigBee CSI is precise in the NLoS scenario. Figure 23(b) illustrates the error of amplitude ratio compared to the ground truth. The error in the amplitude ratio of the estimated ZigBee channel is ≤ 0.052, with the maximum standard variance of 0.07. Since both relative phase and amplitude ratio of the estimated channel are precise, our ZigBee channel estimation is precise in the NLoS scenario.



(a) Relative Phase error.

(b) Amplitude Ratio Error.

Figure 23: The precision of ZigBee channel estimation in the NLOS scenario.



(a) Relative phase error .

(b) Amplitude Ratio Error.

Figure 24: The precision of ZigBee channel estimation in the LOS scenario.

In the LOS scenario, we deploy the ZigBee (USRP) 3 - 21 meters away from X-MIMO and the results are shown in Figure 24. As the distance between ZigBee and X-MIMO increases, the error in the relative phase increases to up 0.11 rad at 21 meters because of the signal strength drop. Despite the SNR drops due to the distance, the error in the amplitude ratio is less than 0.035 and the maximum variance of this error is 0.16. Although the precision of ZigBee channel estimation in the LOS scenario is worse than that in the NLOS scenario, the errors in both phase and amplitude are still very small and negligible.

6.6 Obtaining WiFi-ZigBee Mixed Signal

We manipulate the payload of WiFi fragmented 1 to trigger the ZigBee ACK, which overlaps with the WiFi fragmented 2. In practice, the robustness of the WiFi packet, colliding with ZigBee ACK, depends on the transmission rate of the WiFi packet. On one hand, setting a high transmission rate of WiFi fragments, i.e. applying higher resolution signal emulation, would provide us a higher possibility to successfully trigger the ZigBee ACK. On the other hand, the higher transmission rate indicates the fragment 2 is so vulnerable to the overlapped ZigBee signal that the fragment 2 might be corrupted and the CSI will not be recorded.

To find out the optimal transmission rate of the WiFi fragments, we obtain the rate of successfully triggering ZigBee ACK at the

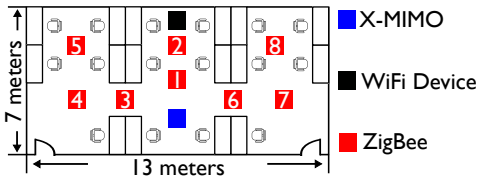
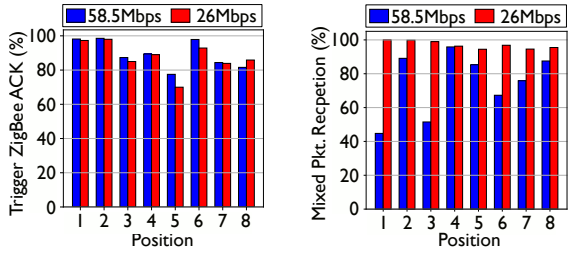


Figure 25: Deployment of ZigBee device, WiFi device (transmits customized WiFi fragments) and X-MIMO in the office.

ZigBee device and packet reception rate of the WiFi fragment 2 at X-MIMO in the settings of different tx rate. Specifically, the ZigBee devices are deployed at eight different positions in the office, as shown in Figure 25. The transmission rates we compare are 58.5 Mbps and 26 Mbps, which apply QAM 64 and QAM 16 mapper to emulate ZigBee. Since the QPSK and BPSK modulations are not suitable to emulate the ZigBee packet with ≥ 50% successful rate, their results are omitted in this section.



(a) Percentage of WiFi fragment 1 successfully triggering ZigBee ACK.

(b) Percentage of WiFi fragments 2 overlapping with ZigBee ACK are successfully received at X-MIMO.

Figure 26: WiFi fragments with 58.5 Mbps and 26 Mbps transmission rate trigger ZigBee ACKs at eight positions.

Figure 26(a) shows the fragmented WiFi packets with 58.5 Mbps tx rate trigger more ZigBee ACKs than 26 Mbps tx rate. This result is expected because the modulation of the 58.5 Mbps WiFi fragment is finer than the modulation of the 26 Mbps WiFi fragment. We also need to notice that, the percentage of successfully triggering ZigBee ACKs by 26 Mbps WiFi fragments is not significantly less than 58.5 Mbps fragments. Even at position 5, the 26 Mbps tx rate achieves a 72% successful rate while achieving more than 80% at other positions.

Figure 26(b) illustrates the percentage of the WiFi-ZigBee overlapped signal to be successfully received at X-MIMO device. As we can see from this result, since the ZigBee devices at positions 1, 3, and 6 are too close to X-MIMO device, the fragment 2 of 58.5 Mbps is easier to be corrupted than the 26 Mbps WiFi fragment. The average success rate of receiving the 26 Mbps WiFi-ZigBee overlapped signal at eight positions is 94.3% while the average success rate of receiving the 58.5 Mbps WiFi-ZigBee overlapped signal at eight positions is only 75.2%. Since the 26 Mbps WiFi packets could emulate ZigBee packets with a high success rate (≥ 80%) and 94.3% of replied ZigBee ACK could be captured in the WiFi CSI, we set the tx rate of WiFi fragments to be 26 Mbps.

6.7 Impact of Transmission Power

We test the impact of transmission power via the symbol error rate at ZigBee devices by controlling transmission power at the X-MIMO device. We deploy X-MIMO device, WiFi device (transmits customized fragments), and ZigBee devices (position 2) in the NLoS scenario, as Figure 15 depicts. We set the transmission power of X-MIMO to be 5 - 17 dBm via `iw` command. Since the two ZigBee devices have a similar symbol error rates, we plot the average SER of two ZigBee devices in Figure 27. When we set the transmission power of X-MIMO to be 11 dBm, the SER (6.8%) is the lowest compared to other settings. Since the default transmission power of the ZigBee device (TelosB mote) is 0 dBm, according to our design in Section 4.3, 8.45 dBm transmission power at each X-MIMO antenna would maintain relative amplitude. Then, the total transmission power of X-MIMO should be 11.45 dBm (=8.45+3). As WiFi hardware only allows us to set the integer transmission power, 11 dBm is the closest legitimate value, leading to the minimum SER.

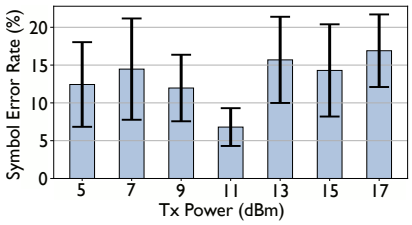
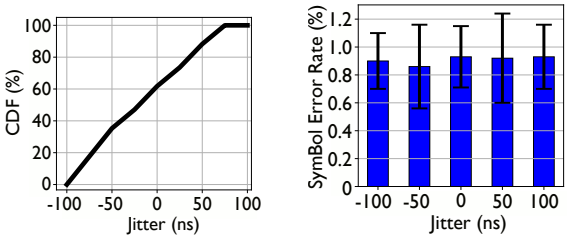


Figure 27: Symbol error rate of X-MIMO under different transmission power.



(a) Cumulative distribution function of ZigBee jitter .

(b) Symbol error rate of X-MIMO suffered from different jitter .

Figure 28: CDF of jitter and corresponding symbol error rate at ZigBee devices.

6.8 Immunity to ZigBee ACK Jitter

In this experiment, we evaluate the impact of the ZigBee ACK jitter. To start, the distribution of the measured jitter is illustrated in Figure 28(a), where the jitter is within the interval of $[-75, 75]$ ns. Then, we use a USRP B210 to precisely control the timing of the WiFi fragments and ZigBee signal, such that the controlled jitter is imposed. By transmitting the customized WiFi and ZigBee signal with different jitter from USRP to X-MIMO device, X-MIMO estimates ZigBee channels and perform cross-technology precoding accordingly.

The symbol error rate of ZigBee suffered from different jitters are plotted in Figure 28(b). The SER is random and ≤ 1% under the jitter from -0.1 μs to 0.1 μs, showing that the ZigBee jitter is negligible in our design. Since the USRP, transmitting customized ZigBee and WiFi fragments, in this experiment does not synchronize its clock to X-MIMO, this result has already involved the influence of CFO and hence no experiment of customizing CFO is conducted. Thus, this SER validates our assertion in Section 4.3 that our cross-technology precoding is immune to the phase uncertainties caused by jitter and CFO.

7 RELATED WORK

MU-MIMO has been studied in many papers. Since they require very precise clock synchronization and precise channel estimation, most of their designs could only be implemented on the software-defined radio [18, 19, 28, 37, 41, 55] or customized hardware [11, 24, 54, 58]. For instance, MURS [19] utilizes an SDR to decode multiple packets simultaneously. Despite Surface MIMO [13] achieves up to 1.3 Gbps throughput on commodity WiFi devices, the design is hard to be applied on low-power devices because (i), the low-power devices cannot support the high-speed signal processing in consideration of energy consumption [20, 59, 60, 67], (ii) low-Power IoT does not support multiple antennas. To improve the spectral efficiency, a few works focus on concurrent communication for IoT [9, 12, 27].

In this paper, we present X-MIMO to enable MU-MIMO from commodity WiFi to commodity ZigBee without any modification of hardware or firmware. X-MIMO explores the channel information in low-power commodity devices, offering more opportunities for wireless sensing and tracking [10, 14–17, 25, 26, 32, 35, 43, 50, 51, 57, 61, 63, 64]. Moreover, X-MIMO extends the single-stream signal emulation to multiple-streams signal emulation, which provides us an opportunity to further push the capacity of WiFi-ZigBee MU-MIMO up to the number of WiFi antennas. These unique features are not supported in existing CTC designs [22, 23, 29–31, 34, 38, 42, 52, 53, 62, 65, 66].

8 CONCLUSION

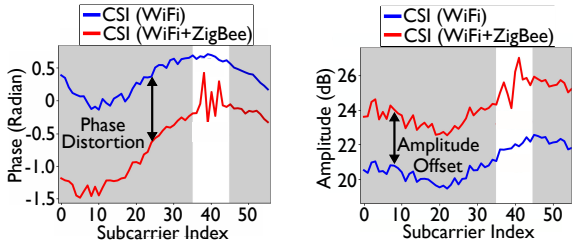
This work presents X-MIMO, a cross-technology MU-MIMO on commodity devices. Utilizing cross-technology channel estimation and precoding, X-MIMO is the first work to offer cross-technology MU-MIMO on commodity devices. Our experiments demonstrate X-MIMO achieves the throughput of 495 Kbps, almost doubling the throughput of legacy ZigBee (250 Kbps), with 99% symbol reliability for two ZigBee receivers. X-MIMO achieves 704.24 Kbps for three ZigBee MU-MIMO and 983 Kbps for four ZigBee receivers (two on each ZigBee channel). Our evaluations also show that X-MIMO performs well in both LOS and NLOS scenarios, where the symbol error rate is $\leq$ 13.6%. Moreover, as the foundation of X-MIMO, cross-technology channel estimation is very precise on commodity devices, offering more opportunities for wireless sensing with low-power IoT devices.

APPENDIX

COMPENSATING HW IMPERFECTIONS

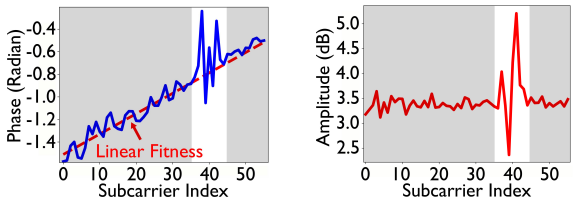
As in Figure 4, computing for H_z assumes that CSIs are similar for the subcarriers that do not overlap with ZigBee (the shaded area), since these two CSIs are measured within the coherence time. However, as shown in Figure 29, it is typically not true. That is, CSI measurements suffer from phase distortion and amplitude offset which must be compensated prior to computing H_z. In other words, the curves in the shaded area need to be matched.

Phase Compensation. Phase distortion stems from the packet boundary detection delay – a jitter in WiFi packet detection time that can be up to sampling duration. This incurs phase shift linearly to the subcarrier frequencies. Let τ be the difference in the boundary



(a) Phase bias. (b) Amplitude offset.

Figure 29: CSIs (with and without overlapped ZigBee) extracted from two WiFi packets within the coherence time do not match due to phase distortion and amplitude offset, which calls for compensation.



(a) Phase Difference. (b) Amplitude Ratio.

Figure 30: Phase difference and amplitude ratio between CSIs of two fragment packets (with and without overlapped ZigBee).

detection time for WiFi fragments 1 and 2. Then, the phase shift for k^{th} subcarrier becomes $2\pi k f_\delta \tau_1$ where f_δ is the subcarrier spacing (=312.5KHz). This causes phase distortion linearly to k (i.e., subcarrier index). As in Figure 30(a) the phase distortion is easily found via linear regression from the phase difference between two CSIs. The compensation would be adding the corresponding phase bias or equivalently, multiplying $e^{j2\pi k f_\delta \tau}$ to the CSI of WiFi fragment 1 for subcarrier k.

Amplitude Compensation. Amplitude offset is caused by Automatic Gain Controller (AGC), a hardware component that dynamically scales the received signal to best fit the ADC range. This leads to an amplitude offset between CSI measurements. Meanwhile, AGC scales all subcarriers in a packet by the same amount. Therefore, as in Figure 30(b), the amplitude offset is simply the ratio between the CSI amplitude, which is consistent among all subcarriers. From this, the amplitude is compensated by multiplying the CSI of WiFi fragment 1 with the amplitude ratio averaged across non-overlapped subcarriers (under gray).

ACKNOWLEDGMENTS

We sincerely thank the anonymous shepherd and reviewers for their valuable comments and suggestions. This work was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-0-01787) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation), the National Research Foundation of Korea (NRF) under grant NRF-2020R1F1A1074657, and the NSF under grant CNS-1717059.

REFERENCES

[1] CC2530 Datasheet. https://www.ti.com/product/CC2530.

[2] IEEE 802.11 Protocol. http://standards.ieee.org/getieee802/download/802.11-2012.pdf.

[3] IEEE 802.15.4 Protocol. http://standards.ieee.org/getieee802/download/802.15.4-2015.pdf.

[4] Implementation of IEEE 802.15.4 Protocol on USRP. https://github.com/bastibl/gr-ieee802-15-4.

[5] WirelessHART, an Industrial Wireless Technology. https://www.emerson.com/en-us/expertise/automation/industrial-internet-things/pervasive-sensing-solutions/wireless-technology.

[6] CC2420 Data Sheet. http://www.ti.com/lit/ds/symlink/cc2420.pdf, 2003.

[7] Isa standard, wireless systems for industrial automation: Process control and related applications. *ISA-100.11 a-2009*, 2009.

[8] Ieee standard for local and metropolitan area networks - part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 4: Alternative physical layer extension to support medical body area network (mban) services operating in the 2360 mhz – 2400 mhz band. *IEEE Std 802.15.4j-2013 (Amendment to IEEE Std 802.15.4-2011 as amended by IEEE Std 802.15.4e-2012, IEEE Std 802.15.4f-2012, and IEEE Std 802.15.4g-2012)*, pages 1–24, 2013.

[9] B. Al Nahas, S. Duquennoy, and O. Landsiedel. Concurrent transmissions for multi-hop bluetooth 5. In *EWSN*, pages 130–141, 2019.

[10] J. Beysens, A. Galisteo, Q. Wang, D. Juara, D. Giustiniano, and S. Pollin. Densevlc: A cell-free massive mimo system with distributed leds. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, pages 320–332, 2018.

[11] A. Bhartia, Y.-C. Chen, L. Qiu, and G. P. Nychis. Embracing distributed mimo in wireless mesh networks. In *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*, pages 66–77. IEEE, 2015.

[12] M. Brachmann, O. Landsiedel, and S. Santini. Concurrent transmissions for communication protocols in the internet of things. In *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, pages 406–414. IEEE, 2016.

[13] J. Chan, A. Wang, V. Iyer, and S. Gollakota. Surface mimo: Using conductive surfaces for mimo between small devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 3–18, 2018.

[14] J. Chauhan, Y. Hu, S. Seneviratne, A. Misra, A. Seneviratne, and Y. Lee. Breathprint: Breathing acoustics-based user authentication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 278–291, 2017.

[15] B. Chen, V. Yenamandra, and K. Srinivasan. Tracking keystrokes using wireless signals. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 31–44, 2015.

[16] Z. Chen, G. Zhu, S. Wang, Y. Xu, J. Xiong, J. Zhao, J. Luo, and X. Wang. m^3: Multipath assisted wi-fi localization with a single access point. *IEEE Transactions on Mobile Computing*, 2019.

[17] J. Ding and R. Chandra. Towards low cost soil sensing using wi-fi. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[18] Y. Du, E. Aryafar, P. Cui, J. Camp, and M. Chiang. Samu: Design and implementation of selectivity-aware mu-mimo for wideband wifi. In *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 229–237. IEEE, 2015.

[19] C. Gao, M. Hessar, K. Chintalapudi, and B. Priyantha. Blind distributed mu-mimo for iot networking over vhf narrowband spectrum. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–17, 2019.

[20] K. Geissdoerfer, R. Jurdak, B. Kusy, and M. Zimmerling. Getting more out of energy-harvesting systems: Energy management under time-varying utility with preact. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 109–120, 2019.

[21] X. Guo, Y. He, J. Zhang, and H. Jiang. Wide: Physical-level ctc via digital emulation. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 49–60. IEEE, 2019.

[22] X. Guo, Y. He, X. Zheng, L. Yu, and O. Gnawali. Zigfi: Harnessing channel state information for cross-technology communication. *IEEE/ACM Transactions on Networking*, 28(1):301–311, 2020.

[23] X. Guo, Y. He, X. Zheng, Z. Yu, and Y. Liu. Lego-fi: Transmitter-transparent ctc with cross-demapping. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2125–2133. IEEE, 2019.

[24] E. Hamed, H. Rahul, and B. Partov. Chorus: truly distributed distributed-mimo. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 461–475, 2018.

[25] M. Hammouda, R. Zheng, and T. N. Davidson. Full-duplex spectrum sensing and access in cognitive radio networks with unknown primary user activities. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2016.

[26] P. Hillyard, A. Luong, A. S. Abrar, N. Patwari, K. Sundar, R. Farney, J. Burch, C. Porucznik, and S. H. Pollard. Experience: Cross-technology radio respiratory monitoring performance study. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 487–496, 2018.

[27] C. Hua, H. Yu, R. Zheng, J. Li, and R. Ni. Online packet dispatching for delay optimal concurrent transmissions in heterogeneous multi-rat networks. *IEEE Transactions on Wireless Communications*, 15(7):5076–5086, 2016.

[28] C. Husmann, G. Georgis, K. Nikitopoulos, and K. Jamieson. Flexcore: massively parallel and flexible processing for large mimo access points. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 197–211, 2017.

[29] V. Iyer, V. Talla, B. Kellogg, S. Gollakota, and J. Smith. Inter-technology backscatter: Towards internet connectivity for implanted devices. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 356–369, 2016.

[30] W. Jiang, S. M. Kim, Z. Li, and T. He. Achieving receiver-side cross-technology communication with cross-decoding. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 639–652, 2018.

[31] W. Jiang, Z. Yin, R. Liu, Z. Li, S. M. Kim, and T. He. Bluebee: a 10,000 x faster cross-technology communication via phy emulation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–13, 2017.

[32] K. Jin, S. Fang, C. Peng, Z. Teng, X. Mao, L. Zhang, and X. Li. Vivisnoop: Someone is snooping your typing without seeing it! In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2017.

[33] V. Jungnickel, K. Manolakis, W. Zirwas, B. Panzner, V. Braun, M. Lossow, M. Sternad, R. Apelfröjd, and T. Svensson. The role of small cells, coordinated multipoint, and massive mimo in 5g. *IEEE communications magazine*, 52(5):44–51, 2014.

[34] S. M. Kim and T. He. Freebee: Cross-technology communication via free side-channel. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 317–330. ACM, 2015.

[35] H. Kong, L. Lu, J. Yu, Y. Chen, L. Kong, and M. Li. Fingerpass: Finger gesture-based continuous user authentication for smart homes using commodity wifi. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 201–210, 2019.

[36] Z. Li and T. He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 2–14, 2017.

[37] K. C.-J. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous mimo networks. *ACM SIGCOMM Computer Communication Review*, 41(4):146–157, 2011.

[38] R. Liu, Z. Yin, W. Jiang, and T. He. Lte2b: time-domain cross-technology emulation under lte constraints. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 179–191, 2019.

[39] H. Lou, M. Ghosh, P. Xia, and R. Olesen. A comparison of implicit and explicit channel feedback methods for mu-mimo wlan systems. In *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 419–424. IEEE, 2013.

[40] A. R. Moghimi, H.-M. Tsai, C. U. Saraydar, and O. K. Tonguz. Characterizing intra-car wireless channels. *IEEE Transactions on Vehicular Technology*, 58(9):5299–5305, 2009.

[41] H. S. Rahul, S. Kumar, and D. Katabi. Jmb: scaling wireless capacity with user demands. *ACM SIGCOMM Computer Communication Review*, 42(4):235–246, 2012.

[42] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang. C-mac: Model-driven concurrent medium access control for wireless sensor networks. In *IEEE INFOCOM 2009*, pages 1845–1853. IEEE, 2009.

[43] L. Shangguan, Z. Zhou, and K. Jamieson. Enabling gesture-based interactions with objects. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 239–251, 2017.

[44] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong. Argos: Practical many-antenna base stations. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 53–64, 2012.

[45] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt. Wirelesshart: Applying wireless technology in real-time industrial process control. In *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 377–386. IEEE, 2008.

[46] P. Sparks. The route to a trillion devices. 2017.

[47] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt. An introduction to the multi-user mimo downlink. *IEEE communications Magazine*, 42(10):60–67, 2004.

[48] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim. Practical mu-mimo user selection on 802.11 ac commodity networks. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 122–134, 2016.

[49] H.-M. Tsai, O. K. Tonguz, C. Saraydar, T. Talty, M. Ames, and A. Macdonald. Zigbee-based intra-car wireless sensor networks: a case study. *IEEE Wireless Communications*, 14(6):67–77, 2007.

[50] G. Wang, C. Qian, K. Cui, H. Ding, H. Cai, W. Xi, J. Han, and J. Zhao. A (near) zero-cost and universal method to combat multipaths for rfid sensing. In *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pages 1–4. IEEE, 2019.

[51] J. Wang, L. Chang, O. Abari, and S. Keshav. Are rfid sensing systems ready for the real world? In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, pages 366—-377, New

York, NY, USA, 2019. Association for Computing Machinery.

[52] S. Wang, S. M. Kim, and T. He. Symbol-level cross-technology communication via payload encoding. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 500–510. IEEE, 2018.

[53] W. Wang, X. Zheng, Y. He, and X. Guo. Adacomm: Tracing channel dynamics for reliable cross-technology communication. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2019.

[54] X. Xie, E. Chai, X. Zhang, K. Sundaresan, A. Khojastepour, and S. Rangarajan. Hekaton: Efficient and practical large-scale mimo. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 304–316, 2015.

[55] X. Xie, X. Zhang, and K. Sundaresan. Adaptive feedback compression for mimo networks. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 477–488, 2013.

[56] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wi-fi. *IEEE Transactions on Mobile Computing*, 18(6):1342–1355, 2018.

[57] M. Yang, L.-X. Chuo, K. Suri, L. Liu, H. Zheng, and H.-S. Kim. ilps: Local positioning system with simultaneous localization and wireless communication. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 379–387. IEEE, 2019.

[58] Q. Yang, X. Li, H. Yao, J. Fang, K. Tan, W. Hu, J. Zhang, and Y. Zhang. Bigstation: enabling scalable real-time signal processingin large mu-mimo systems. *ACM SIGCOMM Computer Communication Review*, 43(4):399–410, 2013.

[59] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. Abdelzaher. Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 278–291, 2018.

[60] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher. Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–14, 2017.

[61] D. Zhang, J. Wang, J. Jang, J. Zhang, and S. Kumar. On the feasibility of wi-fi based material sensing. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[62] J. Zhang, X. Guo, H. Jiang, X. Zheng, and Y. He. Link quality estimation of cross-technology communication.

[63] X. Zhang, D. Yang, L. Shen, X. Chang, J. Huang, and G. Xing. Real-time power profiling of narrowband internet of things networks. In *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos*, pages 90–92, 2019.

[64] M. Zhao, F. Adib, and D. Katabi. Emotion recognition using wireless signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 95–108, 2016.

[65] X. Zheng, Y. He, and X. Guo. Stripcomm: Interference-resilient cross-technology communication in coexisting environments. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 171–179. IEEE, 2018.

[66] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma. Zifi: wireless lan discovery via zigbee interference signatures. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 49–60, 2010.

[67] M. Zimmerling, W. Dargie, and J. M. Reason. Energy-efficient routing in linear wireless sensor networks. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–3. IEEE, 2007.