SDR Receiver Using Commodity WiFi via Physical-Layer Signal Reconstruction

Woojae Jeong, Jinhwan Jung, Yuanda Wang[§], Shuai Wang[†], Seokwon Yang,
Qiben Yan[§], Yung Yi, and Song Min Kim[*]
Korea Advanced Institute of Science and Technology (KAIST)
[§]Michigan State University [†]George Mason University

ABSTRACT

With the explosive increase in wireless devices, physical-layer signal analysis has become critically beneficial across distinctive domains including interference minimization in network planning, security and privacy (e.g., drone and spycam detection), and mobile health with remote sensing. While SDR is known to be highly effective in realizing such services, they are rarely deployed or used by the end-users due to the costly hardware ∼1K USD (e.g., USRP). Low-cost SDRs (e.g., RTL-SDR) are available, but their bandwidth is limited to 2-3 MHz and operation range falls well below 2.4 GHz – the unlicensed band holding majority of the wireless devices. This paper presents SDR-Lite, the first zero-cost, software-only software defined radio (SDR) receiver that empowers commodity WiFi to retrieve the In-phase and Quadrature of an ambient signal. With the full compatibility to pervasively-deployed WiFi infrastructure (without any change to the hardware and firmware), SDR-Lite aims to spread the blessing of SDR receiver functionalities to billions of WiFi users and households to enhance our everyday lives. The key idea of SDR-Lite is to trick WiFi to begin packet reception (i.e., the decoding process) when the packet is absent, so that it accepts ambient signals in the air and outputs corresponding bits. The bits are then *reconstructed* to the original physical-layer waveform, on which diverse SDR applications are performed. Our comprehensive evaluation shows that the reconstructed signal closely reassembles the original ambient signal (>85% correlation). We extensively demonstrate SDR-Lite effectiveness across seven distinctive SDR receiver applications under three representative categories: (i) RF fingerprinting, (ii) spectrum monitoring, and (iii) (ZigBee) decoding. For instance, in security applications of drone and rogue WiFi AP detection, SDR-Lite achieves 99% and 97% accuracy, which is comparable to USRP.

CCS CONCEPTS

• **Networks** → *Cyber-physical networks.*

KEYWORDS

Internet-of-Things (IoT), WiFi, OFDM emulation, Signal reconstruction.

ACM Reference Format:
Woojae Jeong, Jinhwan Jung, Yuanda Wang, Shuai Wang, Seokwon Yang, Qiben Yan, Yung Yi, and Song Min Kim. 2020. SDR Receiver Using Commodity WiFi via Physical-Layer Signal Reconstruction. In *MobiCom 2020 (MobiCom '20), September 21–25, 2020, London, United Kingdom.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3372224.3419189

1 INTRODUCTION

The body of wireless devices has been growing explosively to penetrate into every corner of our living spaces and impact nearly every aspect of our daily lives. This is anticipated to be further intensified with the emergence of the Internet of Things (IoT). Lately, the capability to analyze and diagnose wireless signals has been demonstrated to be critically beneficial in various circumstances including: (i) network management and operation by minimizing interference [6], (ii) security and privacy protection with unauthorized RF activity detection [22], and (iii) mobile healthcare with advanced remote sensing [5].

In highly heterogeneous wireless environment, software defined radio (SDR) is widely considered a de facto solution for signal analysis, by directly accessing the fine-grained physical-layer signal regardless of the underlying wireless technologies – literature have shown its effectiveness through a wide range of applications such as spectrum monitoring for network operation [51], drone detection [49] for security, and activity monitoring for healthcare [15]. Despite the significant benefits, SDRs are rarely adopted in practice and barely used by end-users. This is mainly due to the costly hardware ranging between hundreds to thousands of USD [53]. While low-cost SDRs, e.g., RTL-SDR variants [44–46, 57], are available in the market, the low-end hardware limits their bandwidth to 2-3 MHz and the operation range well below the most popular frequency band of 2.4 GHz unlicensed spectrum (∼1.7 GHz); thus not supporting most of the wireless communication standards (e.g., WiFi, BLE, ZigBee) and a majority of everyday wireless devices such as wearables, earphones, drones, baby monitors, garage openers, and radio controlled cars.

This paper presents SDR-Lite, the first software-only SDR receiver using commodity WiFi, without any additional hardware or modification of firmware. The key idea of SDR-Lite is to trick WiFi to begin packet reception (i.e., the decoding process) when the packet is absent, which is achieved by generating *emulated* packet header – a fake header that does not have a payload. This

[*]Song Min Kim is the corresponding author (songmin@kiast.ac.kr).

MobiCom '20, September 21–25, 2020, London, United Kingdom

ACM ISBN 978-1-4503-7085-1/20/09…$15.00
https://doi.org/10.1145/3372224.3419189

enables commodity WiFi to output the 'decoded' bit sequence corresponding to the ambient wireless signal. SDR-Lite performs *signal reconstruction* on this bit sequence – essentially reverse-engineering the decoding process to closely recover (>85% correlation) the Inphase and Quadrature (I/Q) signals of the original waveform. Thus SDR-Lite imitates SDR receiver, where the reconstructed signal can be facilitated by diverse set of SDR applications.

The effectiveness of SDR-Lite was extensively and comprehensively evaluated by implementing seven distinctive SDR receiver applications under three representative categories: (i) RF fingerprinting, (ii) spectrum monitoring, and (iii) (ZigBee) decoding. Among many encouraging results, SDR-Lite achieves 97% in drone detection (for security), which is comparable with USRP (99.6%). Throughout the applications, SDR-Lite was shown to successfully capture various heterogeneous signals including standard communication signals (BLE and ZigBee), proprietary signals (toy RC car and drone), and non-communication waveform (microwave).

To the best of our knowledge, SDR-Lite is the first to enable SDR receiver functionalities on a commodity WiFi. In particular, SDR-Lite empowers WiFi to effectively capture the physical-layer signal of ambient wireless, thereby spreading the blessing of SDR receiver to billions of WiFi devices and households to enhance our everyday lives. In essence, SDR-Lite redefines the capability of existing WiFi devices and sheds new light on various new directions given the ability of reconstructing fine-grained physical-layer waveform of arbitrary signals. Our contribution is three-fold.

- We design SDR-Lite, a first of its kind, zero-cost SDR receiver using commodity WNIC, without any change in hardware, firmware. This ensures immediate and wide applicability of SDR-Lite on existing WiFi infrastructure.
- SDR-Lite introduces two key techniques: OFDM emulation and signal reconstruction. The former triggers SDR-Lite, where it mimics an OFDM with another OFDM signal, across modulations (e.g., 64 vs. 16 QAM) and standards (802.11n vs. g). The latter rebuilds the signal from the decoded bits.
- We implement SDR-Lite on commodity WiFi devices and USRP B210 for in-depth analysis. SDR-Lite's performance was validated in seven SDR receiver applications across three representative categories of RF fingerprinting, spectrum monitoring, and decoding where we achieved >97% accuracy of drone detection and >94% for WiFi device identification.

2 MOTIVATION

SDR-Lite offers a wide range of benefits ranging from improved IoT to privacy protection.

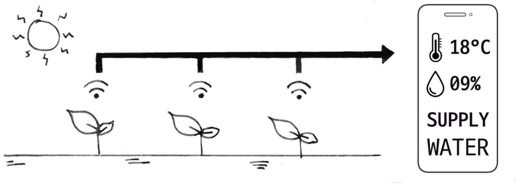
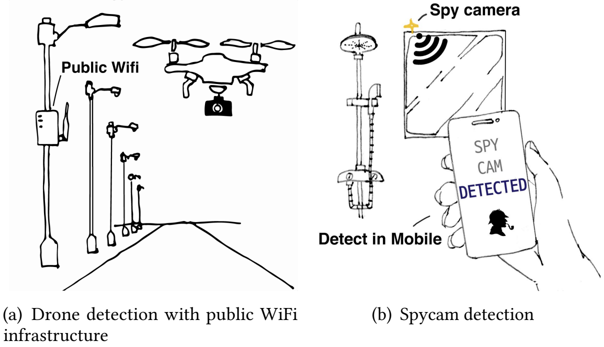


Figure 1: Smart farm data collection

2.1 Mobile IoT Data Collection

A growing number of smart IoT applications and services are enabled by real-time data collected from IoT devices (sensors). ZigBee decoding functionality of SDR-Lite allows to transform WiFi devices (e.g., laptops, mobile phones) into an IoT reader (demonstrated in Section 7.3). Figure 1 illustrates a data collection scenario in a smart farm application. We use a handheld WiFi device (e.g., your mobile phone) to directly read data from the sensors of a smart farm, enabling a real-time and mobile interaction with IoT devices.



(a) Drone detection with public WiFi infrastructure

(b) Spycam detection

Figure 2: Detecting unauthorized devices

2.2 Unauthorized Device Detection

Drone Detection. The growing popularity of these Unmanned Aerial Vehicle (UAV, often called drone) have aroused major security [59, 60] and privacy [13] concerns in private sectors as well as in national defence. SDR-Lite enables a commodity WiFi device to detect and identify a drone's RF signal required to communicate with its controller (demonstrated in Section 7.5). As in Figure 2(a), SDR-Lite turns public WiFi APs, widely deployed in urban areas, into a city-scale drone detection system.

Spycam Detection. Crimes using spycams have been on the rise [47], yet, to effectively detecting these miniature and wireless device is challenging. SDR-Lite can be use to detect various spycams by distinguishing the unique signatures of different RF chips (demonstrated in Section 7.4). As depicted in Figure 2(b), SDR-Lite turns a readily-owned mobile phone into a personal defense system against spycams.

Figure 3: In-home network management

2.3 Network Management

SDR-Lite can be used to monitor the RF spectrum of ISM band (demonstrated in Section 7.2). With a plethora of IoT devices, it is important to plan the wireless network such that the interference is minimized. As depicted in Figure 3, SDR-Lite provides spectrum monitoring functionality to capture various RF signals including non-communication waveforms in 2.4 GHz which serves as a personalized network manager without the expensive SDRs or spectrum analyzers.

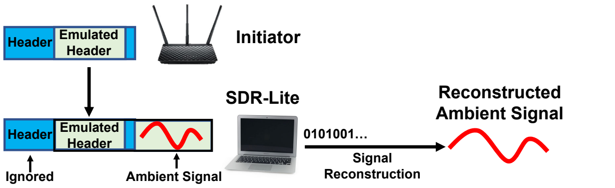


Figure 4: Initiator emulates WiFi header such that SDR-Lite accepts ambient signal as payload and output bits accordingly. SDR-Lite reconstructs the ambient signal from the output bits.

3 SDR-LITE OVERVIEW

Figure 4 illustrates the overall workflow of SDR-Lite. SDR-Lite involves two simple steps. **Step I (Section 4):** SDR-Lite is triggered by a packet from another commodity WiFi, e.g., WiFi Access Point (AP), called *Initiator*. The packet has a short payload only holding *emulated* WiFi packet header – i.e., a combination of payload symbols that closely approximates (i.e., emulates) a real WiFi header (more on emulation in the next section). **Step II (Section 5):** Upon reception of the emulated header, SDR-Lite begins decoding what follows – CRC and ambient signal in the air – and outputs the corresponding bit stream. Finally, SDR-Lite *reconstructs* the ambient physical-layer signal from the bit sequence, which can then be fed into various SDR applications.

We note that the concept of emulation was first proposed in WEBee [39], for cross-technology communication (CTC) from WiFi to ZigBee. In essence, by carefully selecting the payload bytes, the WiFi signal closely imitates the ZigBee signal waveform which is received at the commodity ZigBee device. Inspired by this, we newly introduce OFDM emulation (to build emulated header), which is an emulation between OFDM WiFi variants – e.g., 802.11n and g. This imposes unique challenges including the discrepancies in OFDM symbol structure and constellation points. OFDM emulation is one of our key contributions and primary techniques that enable SDR-Lite.

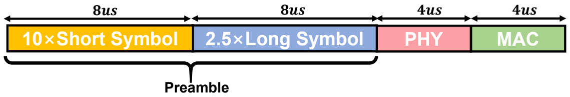
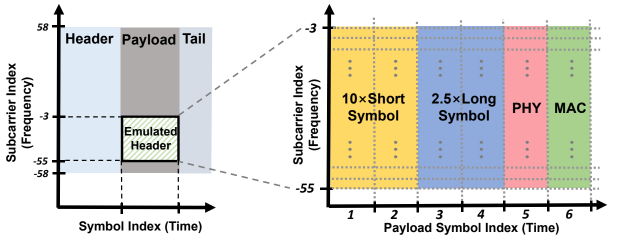


Figure 5: WiFi (802.11g) header structure

4 WIFI HEADER EMULATION

Figure 5 shows the standard-defined WiFi header structure, taking 802.11g as an example. The header consists of three parts – preamble, PHY, and MAC. Preamble is used for packet detection, synchronization, and channel estimation. PHY header carries physical-layer information, i.e., modulation type, code rate, and packet length. MAC header holds link-layer specifics such as protocol version and packet type. Upon receiving a WiFi header, a receiver begins decoding the payload following the header, for the duration of packet length, using the modulation and code rate specified in the header.

Initiator emulates the entire header (preamble, PHY, MAC header), which triggers the decoding process upon reception. However, as the emulated header sits within the payload, the decoding is instead applied to the ambient signal that follows the packet. This yields output bit sequence, from which the received ambient signal can be reconstructed. We note that the reception of emulated header requires bypassing the original WiFi header that always precedes the emulated header in the payload; otherwise, emulated header will simply be decoded as payload. Below we first discuss how to bypass the original header, followed by the details on generating emulated header.



(a) Initiator (40 MHz 802.11n) (b) Emulated Header (20 MHz 802.11g)

Figure 6: OFDM Emulation

4.1 Bypassing the Original Header

In essence, SDR-Lite bypasses the original header in the Initiator's packet while accepting the emulated header, when their center frequencies are different. In other words, the Initiator generates an emulated header with the center frequency deviating from its own. One practical way to achieve this is to set the Initiator to 40 MHz-bandwidth 802.11n, which is commonly supported in typical WNICs on the market. Among the 40 MHz, a partial spectrum of 20 MHz is used to make the emulated header, where the center frequency differs from the Initiator. Figure 6(a) demonstrates an example setting used in our implementation where the emulated header uses 52 subcarriers (=16.25 MHz, except guard band), ranging [-55,-3]. The center frequency of the emulated header, which is 2.417 GHz (i.e., WiFi channel 2), is clearly apart from that of the entire 40 MHz. Therefore, SDR-Lite operating at WiFi channel 2 receives the emulated header while bypassing the original header. SDR-Lite on other channels may as well be supported by varying Initiator channel and subcarrier allocation for emulation.

4.2 OFDM Emulation

Emulated header lies within the payload of the Initiator packet. SDR-Lite aims to minimize emulation errors under WiFi hardware constraints, including the discrepancy in the symbol structure and constellation points (due to the disparate modulations) between header and the payload. We address the discrepancy in symbol structures via *symbol mapping* and disparate constellation points through *subcarrier constellation mapping*.

4.2.1 Symbol Mapping. Figure 6(b), in the x-axis, illustrates how each part of the header symbols are mapped to (i.e., emulated by) payload symbols. As the figure shows, PHY and MAC are one-to-one mapped to (and emulated with) one payload symbol, depicted as symbol 5 and 6. This is because PHY and MAC headers share the same symbol structure as the payload – i.e., 3.2 μs symbol plus 0.8 μs cyclic prefix (CP). However, the preamble symbols structures significantly differ from the payload. For instance, short and long symbols have duration of 0.8 μs and 3.2 μs, respectively, both of which do not have CP. To tackle this, we provide in-depth discussion on mapping preamble symbols to payload symbols, essentially enabling emulation under symbol structure discrepancy.

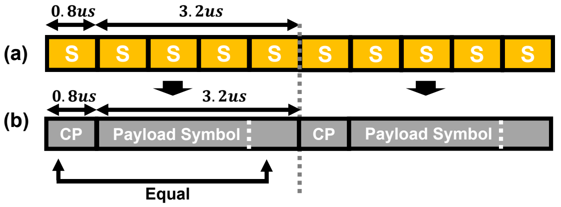


Figure 7: (a) 5 × short symbols (=S) can be emulated with (b) a payload symbol + CP

Short Symbol ↔ Payload Symbol Mapping. As Figure 7 illustrates, a short symbol duration is only 0.8 μs while payload symbol lasts for 3.2 μs. Moreover, a payload symbol is prepended with a 0.8 μs CP, which is a copy of the last 0.8 μs of payload symbol (to avoid inter-symbol interference). Meanwhile, short symbol does not have CP. Under the discrepancy lies a hidden emulation opportunity by considering short symbols in a batch of five. This is shown in Figure 7: payload symbol (3.2 μs) emulates 2^{nd}-5^{th} short symbols (4x0.8=3.2 μs). By emulating 5 short symbols, the CP requirement is inherently satisfied, since the first short symbol (at the place of CP) is equivalent to the 5th short symbol. A WiFi preamble contains 10 short symbols, which can be emulated with two payload symbols.

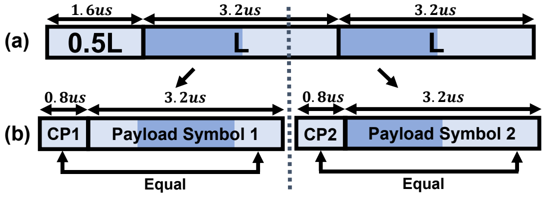


Figure 8: (a) 2½ long symbol (=L) sequence in the preamble (b) Mapping to two payload symbols for emulation.

Long Symbol ↔ Payload Symbol Mapping. As in Figure 8(a), there are 2½ long symbols in the preamble. The preceding ½ long symbol matches with the latter ½ of the following long symbols, represented in lighter blue, while the former half is in darker blue. Figure 8(b) clearly shows that 2½ long symbol sequence can be represented by two payload symbols. Moreover, for each payload symbol, the last 0.8 μs matches the initial 0.8 μs, which complies with payload CP requirement. This demonstrates that 2½ long symbol sequence can also be effectively emulated with payload symbols.

4.2.2 *Subcarrier Constellation Mapping.*

Given the mapped symbols, we turn to emulating subcarriers via constellation mapping. As an example, let SDR-Lite be 802.11g: the preamble and PHY are modulated in BPSK, while MAC uses 16 QAM. On the contrary, payload has a wide selection of modulation options depending on Modulation and Coding Scheme (MCS). Among them we adopt 64 QAM, the constellation with the finest granularity, to minimize error in constellation mapping; thus, the Initiator is configured to 64 QAM of code rate 5/6 (802.11n MCS 7).

Preamble/PHY Header (BPSK). Figure 9(a) illustrates the preamble (long + short symbol) and PHY header's BPSK constellation points in relation to the 64 QAM points used in the payload. To minimize the emulation error, 64 QAM constellation points closest to BPSK should be selected for emulation, under the restriction imposed by the code rate. Figure 9(b) illustrates the emulation of short symbol; 5/6 code rate allows to select only up to 4 bits out of 6 bits assigned per 64 QAM point, where the remaining 2 bits are uncontrollable (generated by coding). Under this constraint, we achieve the minimum emulation error by selecting the 4 bits in common in the four closest points (in blue). This ensures emulation to one of the four closest points. We note this mechanism effectively leverages an intrinsic feature of the 64 QAM points where the adjacent points differ by only 1 bit (i.e., gray coding). The same idea applies when emulating long symbol and PHY. As in Figure 9(c), selecting the four common bits (in red) among the four closest points ensures emulation to one of them, which minimizes the emulation error.

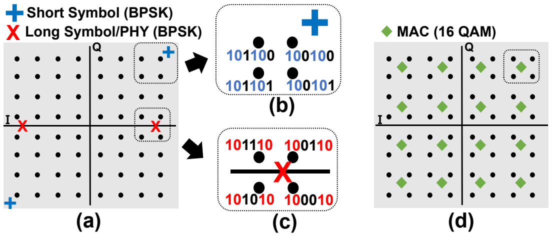


Figure 9: Subcarrier constellation mapping minimizes the error under constellation discrepancy between header and payload symbols

MAC Header (16 QAM). The dotted box in the Figure 9(d) depicts the constellation mapping of 16 QAM used in the MAC header. This follows the same mechanism as in BPSK, where four closest 64 QAM points are mapped to each 16 QAM point. Like BPSK constellation mapping, this also effectively minimizes the emulation error under practical WiFi hardware constraints.

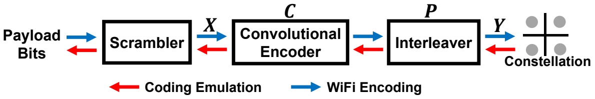


Figure 10: Coding emulation is a reverse of WiFi encoding.

4.3 Coding Emulation

The channel coding of WiFi introduces redundancy, which imposes challenges for emulation in a reverse order [39]. Figure 10 depicts the encoding mechanism for WiFi in blue arrows, which is processed per-symbol basis. We note that the OFDM emulation outputs 6 coded bits per subcarrier (corresponding to a 64 QAM point). One symbol of the Initiator, set to 40 MHz 64 QAM 802.11n, has 108 data subcarriers. This translates to a total of 648 (=108×6 bits/subcarrier) coded bits, denoted Y. Coding emulation, shown as the red arrow in Figure 10, is to find the pre-coded bit sequence, X, that yields Y via WiFi encoding (blue arrows). In other words, we compute X for a given Y – essentially reverse-engineering the interleaver and the convolutional coding.

The coded bits of Y is generated from 540 (=648x5/6 code rate) input bits of X plus 6 bits carried over from the previous symbol, X_b. Let Galois Finite field matrices (GF(2)) of P and $[C_b\ C]$ represent the interleaving and convolutional coding matrices, respectively. C_b is the first 6 columns of the convolutional matrix convoluted X_b. Then, the WiFi encoding is formulated as:

$$P[C_b\ C]\begin{bmatrix}X_b\\X\end{bmatrix} = PC_bX_b + PCX = Y, \qquad (1)$$

where PC_bX_b is a constant, since X_b is given from the previous symbol while C_b and P are both fixed by the WiFi standard. Then, we let $Y' = Y - PC_bX_b$ and reformulate Eq. (1) into a linear equation:

$$(PC)X = Y', \tag{2}$$

PC is 648x540 matrix. To emulate 54 subcarriers (20 MHz) each with 4 selected bits (as per the subcarrier constellation mapping), a total of 216 are chosen among 648 output bits in Y. Let the 216 bit subvector be Y_{216} and corresponding PC matrix be $(PC)_{216}$. Then, $(PC)_{216}X = Y_{216}$. Finally, $(PC)_{216}$ matrix is full rank based on the standard [64]. Given Y, we get X considering the interleaver and convolutional encoder.

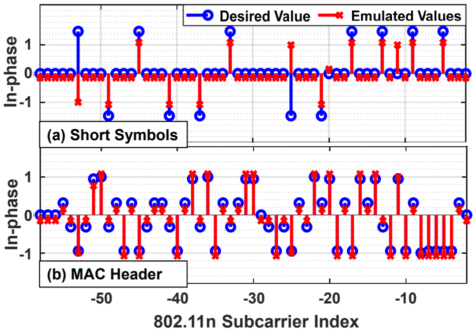


Figure 11: An example of emulated header in frequency domain

Figure 11 demonstrates an example of the Initiator emulating WiFi header, which is represented in the frequency domain. Figure 11 (a) illustrates the Short symbol (BPSK), where long symbol and PHY are similar as they are also in BPSK. MAC (16 QAM) is shown in Figure 11 (b). While largely similar, the difference between the two signals indicates the emulation error from disparate constellation points. It is worth noting that the payload bits can be computed from X by reversing the scrambling block, which is a straightforward process (here we omit for brevity) as the scrambling is simply an XOR operation with a given bit sequence of scrambling seed. The scrambling seed can be easily tracked on commodity WNICs – e.g., widely used Atheros WNIC (e.g., AR9380) increases the seed by one for every packet transmission [32].

4.4 Emulation Imperfections

Uncontrollable Pilot Subcarriers. As per WiFi standard, the three pilot subcarriers in WiFi have a set of preassigned values for any given symbol index. This is enforced by the hardware and thus cannot be altered. That is, pilot subcarrier values are uncontrollable which may incur errors if they do not match the emulation. We minimize this impact by carefully selecting the symbol index from which the Header emulation begins. Specifically, we exploit the sequence of pilot values that best match the emulated value. We find that, by beginning the header emulation from the 3^{rd} symbol, the preassigned values for three pilot subcarriers closely matches the corresponding subcarrier values in the emulated header. This matches 8 out of 9 values in the emulated header, where, from our experiment, the impact of a single mismatch was negligible. To conclude, impact of pilots are minimized by simply emulating the header beginning from the 3^{rd} payload symbol.

Impact on Equalization. WiFi estimates the channel from the long symbol, which is used for the equalization in decoding. That is, slight emulation error due to constellation discrepancy in the long symbol leads to imperfect channel estimation and equalization, thereby lowering SNR. We experimentally evaluate the impact of emulation imperfections including uncontrollable pilot subcarriers and equalization in Section 7.

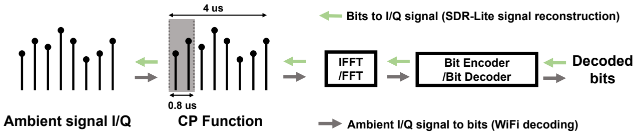

4 us
Bits to I/Q signal (SDR-Lite signal reconstruction)
IFFT /FFT
Bit Encoder /Bit Decoder
Decoded bits
0.8 us
Ambient signal I/Q
CP Function
Ambient I/Q signal to bits (WiFi decoding)

Figure 12: Ambient signal reconstruction.

5 SIGNAL RECONSTRUCTION AND APPS

We discuss reconstructing the ambient I/Q signals from the received bit streams and signal processing for SDR receiver applications.

5.1 Ambient Signal Reconstruction

Upon receiving the emulated header, SDR-Lite begins receiving (i.e., decoding) ambient signals for the duration of LENGTH (defined in PHY). This process is drawn in Figure 12 as grey arrows, where it outputs corresponding decoded bit sequences. We note that this is the standardized WiFi (e.g., 802.11g) decoding which is automatically processed on the WNIC – CP removal, FFT, and decoder including interleaver, convolutional decoder, and scrambler. Ambient signal reconstruction is a process of reversing the decoded bit sequence back to the original ambient I/Q signal, on which SDR applications are performed. This is depicted in green arrows in Figure 12, which essentially follows the WiFi encoding mechanism as we discuss the details in the following.

Signal reconstruction first performs bit encoding, which includes scrambling, convolutional encoding, and interleaving. The interleaving is a fixed operation defined in the WiFi standard. Scrambling and convolutional encoding take the scrambling seed and the coding rate as the parameter, respectively. We note that both the scrambling seed and coding rate are controlled by the OFDM emulation (thus known). Therefore, scrambling and convolutional encoding operations are also determined. The bit sequence obtained from the bit encoding are mapped to the constellation points (e.g., 64 QAM) to yield signal I/Q values. These I/Q signals are in frequency domain (allocated to subcarriers) from which the time domain signal is reconstructed through the inverse FFT. This outputs 3.2 μs time domain signal, which corresponds to a WiFi symbol without CP. Lastly, CP is prepended at the beginning of the symbol by copying the last 0.8 μs. This recontructs the ambient signal for the WiFi symbol duration of 4 μs. This process is repeatedly performed for the duration of LENGTH in the emulated header.

Figure 13 demonstrates an example of a reconstructed signal. It has inevitable errors and phase rotation introduced by the WiFi hardware. In the following we discuss the three different types of errors (CP, boundary, and convolutional error) as well as how to selectively leverage the signal portion (i.e., white box in Figure 13), so as to maximize the correlation between the original and the reconstructed signal. This is validated to be highly effective in our evaluations. For instance, SDR-Lite achieves <5% symbol error rate for ZigBee decoding.

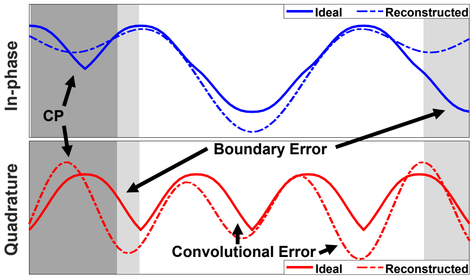


Figure 13: Reconstructed signal and errors.

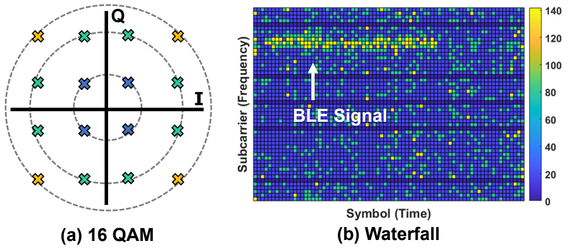
CP and Boundary Error. The first and most significant source of error is the CP, indicated as dark gray in Figure 13. The CP portion (the first 0.8 μs) is overwritten by the last 0.8 μs of the symbol, and thus the original ambient signal at the place of CP is entirely lost. The second is the boundary error in light gray in Figure 13. This occurs due to the cyclic property of the Discrete Fourier Transform (DFT) which necessitates the beginning and end of the symbol to be always identical, deviating the symbol boundaries of the reconstructed signal from the original signal. If we let reconstructed signal be $x[n] = \sum_k X[k]e^{j2\pi kn/T}$, $x[0] = x[T]$ where T is the symbol duration. Moreover, the boundary error is inversely proportional to the bandwidth. Intuitively, this is because higher frequency signal better allows abrupt changes in a shorter duration (thus smaller boundary error). To sum up, let f_H be the highest positive frequency component of the signal (e.g., 1 MHz for ZigBee), we set the boundary error plus CP to $1.5/f_H$ (= 1.5 μs in ZigBee) based on our experiment, which achieves 0.86 correlation between ideal and recovered signals.

Convolutional Error. As the convolutional encoding limits the degree of freedom, WiFi is unable to represent all kinds of ambient signals. In other words, if the ambient signal do not fit into any WiFi code word, it is decoded to the code word with the lowest hamming distance from the ambient signal. As in Figure 13, this induces the convolutional error between the reconstructed signal and the original ambient signal waveform. SDR-Lite effectively suppresses the convolutional error by leveraging the unique features of Viterbi decoding and Gray coding in WiFi. That is, the maximum likelihood property of the Viterbi decoding optimally selects the code word with the lowest hamming distance. By Gray coding, this minimum bit difference indicates closeness in the constellation point, or similarity to the original ambient signal.

The reconstructed signal is also phase-rotated, due to the ambient noise at the pilot subcarrier. This is because WiFi combats phase error by compensating the phase difference between the received pilot subcarrier and the reference value. In SDR-Lite, pilot subcarriers are populated by the ambient noise, causing a random phase rotation to the reconstructed symbol. We note that the phase rotation causes issues in some application (e.g., decoding) and but not others (e.g., RF fingerprinting and spectrum monitoring). Next, we discuss further processing of the reconstructed signal depending on the applications.

5.2 Application I: RF Fingerprinting

The reconstructed signals closely approximate the original signals, thereby keeping the unique physical-layer signature intact. This indicates that the reconstructed signal can be used for RF fingerprinting. Despite the random symbol phase rotation, an entire symbol is rotated by an fixed phase shift, keeping the relative phases between the subcarriers consistent with the original signal. We note that the amplitude is unaffected by the phase rotation. In other words, the amplitude and relative phases among subcarriers of the emulated signal directly reflect the physical-layer signature of the original signal. Capitalizing on this property, SDR-Lite enables various RF fingerprinting applications using commodity WiFi. Section 7 demonstrates these applications including smartphone identification, and rogue WiFi AP detection.



(a) 16 QAM (b) Waterfall

Figure 14: Waterfall from signal. Can be used for Spectrum analysis.

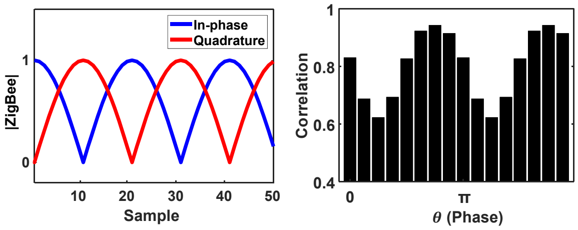
5.3 Application II: Spectrum Monitoring

Spectrum monitoring investigates diverse activities on the wireless channel via fine-grained power measurements. Since the power is unaffected by the phase rotation, the reconstructed signals can be directly used without further processing. Here, the precision of the power measurement largely depends on the modulation of the reconstructed signal. Figure 14(a) illustrates an example of 16 QAM that yields a 3-level precision (blue, green, yellow) based on constellation points. We note that the precision can be enhanced simply by adopting a higher order QAM (e.g., 10 levels for 64 QAM). Figure 14(b) shows a waterfall plot obtained from 16 QAM reconstructed signal capturing a 0.192 ms-long signal beginning at the 4^{th} symbol. From the subcarrier index on which the signal is residing, this signal is identified as a BLE with the central frequency and bandwidth of approximately 2.422 GHz (=BLE channel 9) and 1 MHz, respectively. In particular, SDR-Lite offers spectrum monitoring functionality with timing and frequency precision of 4 μs and 312.5 KHz, corresponding to symbol duration and subcarrier spacing, respectively.

The duration of the monitoring (i.e., the length of the reconstructed signal) is determined by the 12-bit LENGTH field (in 802.11g) in the emulated header. This allows SDR-Lite to freely set the LENGTH to the maximum of 4095 ($2^{12} - 1$) Bytes beyond the MTU (2304 Bytes), indicating the monitoring duration of 950 μs (16 QAM) - 1.86 ms (QPSK) depending on the modulation. We experimentally validated the 4095 Bytes LENGTH is successfully received on commodity WNIC [2]. Also, it is non-disruptive to the existing networks given that the WiFi supports long aggregated packets of A-MPDU 65535 Bytes [64]. To capture a longer signal, consecutive emulated headers can be leveraged. Given the average channel access delay of 101.5 μs [41] and the 950 μs monitoring duration per emulated header. This corresponds to monitoring 89.4% of the time.

5.4 Application III: Decoding

The spectrum monitoring identifies the wireless signal characteristics including the technology, central frequency, duration, and arrival time. Given such knowledge, we can further decode the signals. We take ZigBee as an example to illustrate the decoding mechanism.



(a) Reference ZigBee signal (b) Correlation vs. θ

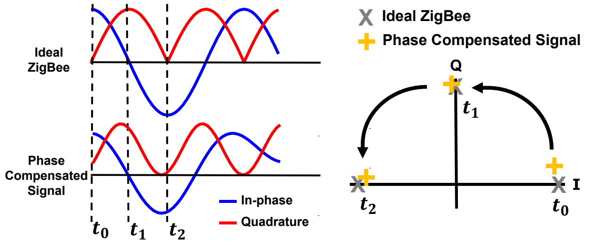
Figure 15: Phase compensation

Decoding a phase modulated signal, such as ZigBee, requires compensating the phase rotation. To achieve phase compensation, we exploit a common signal feature inherent in arbitrary ZigBee signals, regardless of the data they carry (which is unknown before decoding). Due to the +/- half-sine shaped ZigBee signal, the absolute value (i.e., taking $|\cdot|$ for both in-phase and quadrature) of an arbitrary ZigBee signal invariably becomes Figure 15(a). We leverage this signal as a reference to phase compensation – We first take the absolute value of the reconstructed signal followed by phase rotation of θ, which is then correlated to the reference signal in Figure 15(a) for different θ such that the correlation is maximized. That is:

$$\underset{\theta}{\operatorname{argmax}} \ \operatorname{Corr}(R(t), |\mathrm{I}\{x(t)e^{j\theta}\}| + j|\mathrm{Q}\{x(t)e^{j\theta}\}|), \tag{3}$$

where $R(t)$ and $x(t)$ are the reference and phase rotated signal, respectively, and we increase θ by $\pi/8$ in our implementation. This compensates the symbol phase rotation of $-\theta$. Figure 15(b) demonstrates an example of the correlation with respect to θ where it reaches the peak correlation of 0.93 at $\frac{3}{4}\pi$ and $\pi + \frac{3}{4}\pi$. This is because the absolute value of a signal and its π-shifted version is equal. That is, $|\mathrm{I}\{x(-t)e^{j\theta}\}| = |\mathrm{I}\{x(t)e^{j\theta}\}|$, which holds for quadrature as well.

Selecting between the two peaks leverages the direct sequence spread spectrum (DSSS) in ZigBee. That is, DSSS predefines a set of signal sequence, which is met strictly by only one of the two candidates (i.e., $\frac{3}{4}\pi$ and $\pi+\frac{3}{4}\pi$). This compensates the symbol phase rotation.



(a) Time Domain Signal (b) Constellation

Figure 16: (a) Ideal ZigBee (upper) vs. phase compensated (reconstructed) ZigBee (lower) (b) Both yields the same bits of two consecutive 1s for $t_0 \rightarrow t_1$ and $t_1 \rightarrow t_2$.

Figure 16(a) depicts phase compensated signal compared to the ideal signal, followed by the decoded result in Figure 16(b). In ZigBee, a clockwise or counter-clockwise phase shift indicates 1 or 0, where the phase compensated signal decodes to correct bits (i.e., two consecutive ones). We also note that mapping to the DSSS signal sequence simultaneously addresses the impact of CP and boundary error, by essentially filling in the signal loss. Therefore, the signal is fully decoded. We rigorously evaluate the ZigBee decoding performance in Section 7.3.

6 DISCUSSION

This section discusses the equalization impact and WNIC settings of SDR-Lite.

The Impact of Equalization Due to the channel between the Initiator and SDR-Lite, SDR-Lite will automatically equalize received signal including ambient signal by using channel state information (CSI). This affects the reconstructed signal from SDR-Lite. Nevertheless, in case of reconstructing narrowband signal (e.g., ZigBee), the impact of equalization is minor. That is, for narrowband signal, the channel is flat such that every frequency component of reconstructed signal is consistently affected by flat channel. Thus, the entire signal's magnitude and phase are affected in a consistent manner. This effect could be mitigated through our design in Section 5.4 with phase correction. In case for the wideband signal, we could unequalize the reconstructed signal by using CSI, which can be obtained by sending general 802.11n packet to SDR-Lite using tools, i.e., Atheros CSI tool [66].

WNIC Settings. OFDM emulation uses consecutive 53 subcarriers out of 128 in 40 MHz 802.11n, while not containing null subcarriers. Our implementation uses subcarriers [-55, -3] under which the center frequency of Initiator is 9 MHz apart from SDR-Lite. This is compatible to the commodity WiFi, as typical WNICs allow center frequency shifts in steps of 1 MHz at the driver level. Also, to enable reception of the ambient signals which are naturally random, SDR-Lite disables the CRC as allowed in many commodity WNICS [3, 4, 14, 17]. Under these settings, SDR-Lite has potential to be extended to other 802.11 variants with 40 MHz bandwidth option (e.g., 802.11ac).

7 EVALUATION

In this section, we perform extensive experiments to evaluate SDR-Lite under various circumstances. The experimental testbed consists of an Initiator and SDR-Lite where the Initiator sends 40 MHz WiFi packets (including emulated header) to the 2.426 GHz which is 1 MHz shifted from the WiFi channel 4, while SDR-Lite receives the emulated header at 2.417 GHz (the WiFi channel 2). We use a commodity WNIC Atheros AR9380 and USRP B210 as the Initiator, and D-Link DWA-192 and Alfa AWUS036ACM as SDR-Lite. USRP B210 is used for detailed analysis.

7.1 Basic Performance of SDR-Lite

In this subsection, we evaluate basic performance of SDR-Lite in terms of the shape of emulated header and packet reception rate (PRR) performance with varying distances between the Initiator and SDR-Lite. We use AR9380 (WNIC) as the Initiator and Alfa AWUS036ACM (WNIC) as SDR-Lite, respectively.

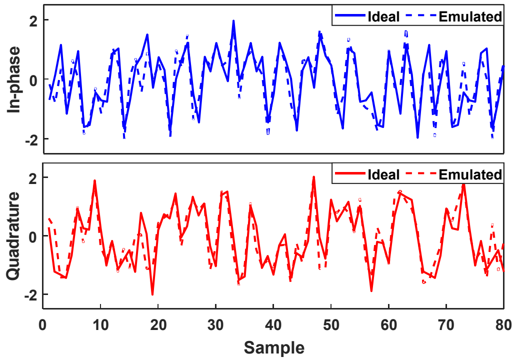
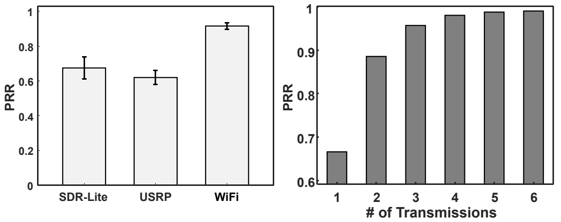


Figure 17: Emulated signal vs. ideal signal in the time domain.

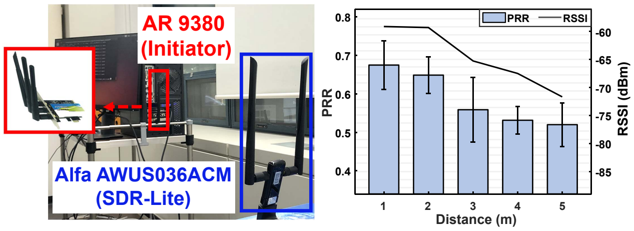
Emulated Header. Figure 17 illustrates the emulated header in comparison to the ideal signal in the time domain. Despite emulation imperfections (discussed in Section 4.4), emulated signal closely mimics the ideal signal with the signal-to-noise ratio of 6.02 dB. This clearly demonstrates the practicality of the OFDM emulation and its compatibility to the commodity WiFi receivers.



(a) PRR under three settings (b) PRR vs. # of Transmissions

Figure 18: PRR of emulated header.

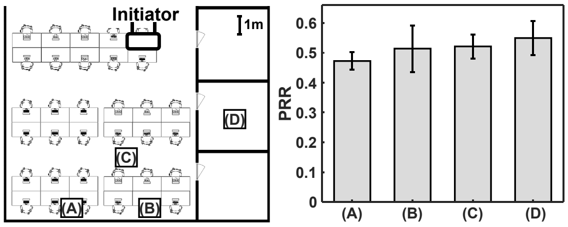
SDR-Lite: Emulated Header PRR. In order to evaluate PRR, we let the Initiator send a 40 MHz 802.11n packet which includes a 802.11g emulated header with 16 QAM and code rate 3/4 in the payload with the transmit power (TX power) 18 dBm. In this evaluation, we choose AR 9380 and USRP as the Initiator that transmits the emulated header embedded in the 40 MHz 802.11n packet and let SDR-Lite receive the emulated header which is 2 m apart from the Initiator. Figure 18(a) shows that the PRRs achieved by the commercial WNIC and USRP are 67% and 62%, respectively. We note that the USRP's PRR is lower than that of the WNIC since it does not perform CSMA/CA. The PRR for normal (i.e., non-emulated) WiFi packet is 97.79 %. We evaluate the retransmission scenario for higher reliability. Figure 18(b) demonstrates SDR-Lite achieves packet reception of above 99% with three retransmissions (i.e., total of four transmissions).



(a) LoS Scenario (b) Packet Reception Ratio

Figure 19: PRR vs. distance in the LoS.

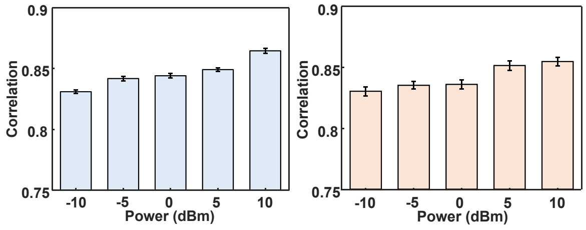
Emulated Header: Line-of-Sight. To validate the performance of SDR-Lite under various scenarios, we consider both Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS) for PRR evaluation. First, Figure 19(a) depicts the experimental setup of the LoS scenario. We measure the PRR and RSSI as the distance between the Initiator and SDR-Lite increases with the TX power 18 dBm. As shown in Figure 19(b), at the closest distance of 1 m the PRR is 67% with -59 dBm, and it drops to 52% at 10 m with RSSI degraded to -72 dBm. However, we note that by retransmitting emulated headers five times, PRR reaches up to 95% at 10 m.



(a) NLoS Scenario (b) Packet Reception Ratio

Figure 20: PRR in the NLoS environment.

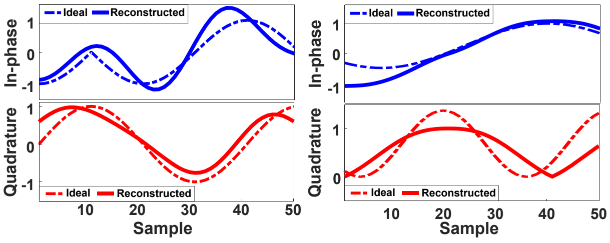
Emulated Header: Non-Line-of-Sight. The impact of NLoS environment is also evaluated. Figure 20(a) shows various SDR-Lite positions (A)-(D) in the NLoS environment where the Initiator is fixed on the desk which can be regarded as a WiFi AP. As shown in Figure 20(b), the PRRs at (C) and (D) are 52% and 55%, respectively, while in case of (A) and (B), the PRRs become 47%, 52%, respectively. This is because the positions of (A) and (B) are further away from the Initiator than the other positions. We note that by retransmitting the emulated header five times, all users in the NLoS scenario can achieve 95% of PRR.



(a) ZigBee correlation (b) BLE correlation

Figure 21: Reconstructed and ideal signal correlation.

Signal Reconstruction. Now we demonstrate that non-WiFi signals (i.e., ZigBee and BLE) can be reconstructed by SDR-Lite. We record emulated header signals (sent by AR 9380) and ZigBee signals (sent by CC2650) using a USRP, and then the concatenated signals are transmitted by USRP to SDR-Lite for convenience. Figure 21 demonstrates the maximum correlation value θ between the reconstructed ZigBee (or BLE) signal and the ideal ZigBee (or BLE) signal, when the Tx power varies from -10 dBm to 10 dBm. As shown in Figures 21(a) and 21(b), the maximum correlation value can reach 0.86 and 0.85 for ZigBee and BLE, respectively, when Signal to Noise Ratio (SNR) is sufficiently high. We note that the achieved correlation indicates that the received signal (which is non-WiFi) by SDR-Lite can be successfully reconstructed to recover the original signal as illustrated below.



(a) Phase Corrected ZigBee (b) Phase Corrected BLE

Figure 22: Reconstructed ZigBee and BLE.

Figures 22(a) and 22(b) show the phase corrected signals of ZigBee and BLE through the phase compensation with θ as described in Section 5.4. As illustrated, the reconstructed signal (solid line) and the ideal signal (dashed line) show high similarity, so that the original payload can be decoded from the reconstructed signal (See Section 7.3).

7.2 Spectrum Analysis

In this subsection, we demonstrate an application of SDR-Lite as a spectrum analyzer. Using the Initiator as AR 9380 or USRP, SDR-Lite (D-Link DWA-192) can monitor the RF spectrum in the ISM band. As a ZigBee/BLE device, we use CC2650 [58] which is multi-standard.

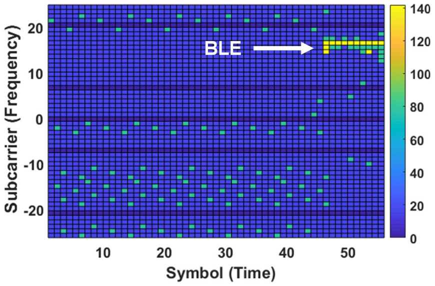


Figure 23: BLE captured by SDR-Lite.

How to Monitor Spectrum. To analyze the RF spectrum, we show that for various RF activities SDR-Lite is capable of measuring power of RF signals (thus plotting waterfall). Upon receiving the emulated header sent by the Initiator, SDR-Lite starts to monitor its RF spectrum at 2.417 GHz with 20 MHz bandwidth. Figure 23 shows that SDR-Lite captures BLE signal sent at 2.422 GHz where X axis represents the time domain with a unit of a WiFi symbol (4 μs) and Y axis represents the frequency domain with a unit of a subcarrier (0.3125 kHz).

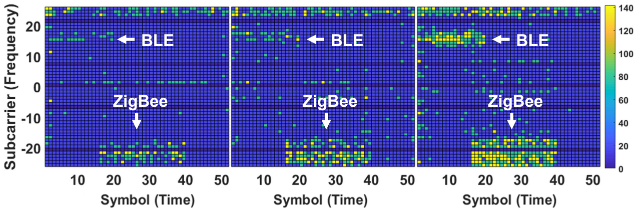
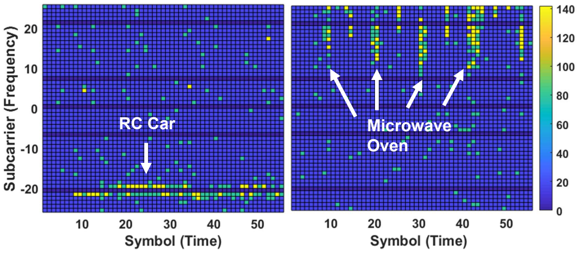


Figure 24: Waterfall under varying power levels.

Spectrum Analysis: Emulated Header Power. The measured RF power by SDR-Lite depends on emulated header's power. To evaluate the impact of the power of emulated header, we record emulated header and ZigBee/BLE signals to USRP, and then send the signals to SDR-Lite where the power ratio of the emulated header to BLE/ZigBee varies from 16:1 to 1.6:1. Figure 24 demonstrates the waterfall plots that capture the ZigBee and BLE signals at the center frequency of 2.410 GHz and 2.422 GHz, respectively. As the power of the emulated header decreases (from the left to the right), the plot shows more detailed power levels of the ZigBee and BLE signals. When an RF signal is received by SDR-Lite, it is required to control the power of the emulated header close to the RF signal so as to map the power of RF signal to the proper QAM point.

Figure 25: RC car and microwave oven.



(a) Radio Controlled Car **(b) Microwave Oven**

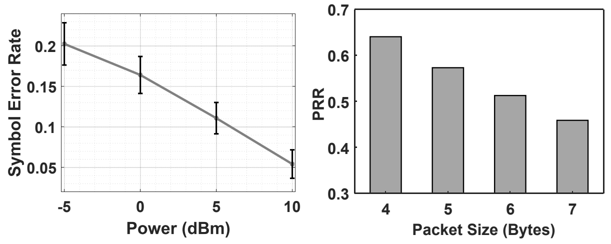
Figure 26: Proprietary and non-communications signals.

Spectrum Analysis: Other RF Signals. SDR-Lite is also able to monitor non-wireless protocol signals. With the same experimental setup, we capture RF signals from the radio controlled (RC) car and the microwave oven by SDR-Lite. Figures 26(a) and 26(b) show the waterfall plots of the RF signals from the RC car and microwave oven, respectively. It demonstrates that SDR-Lite can monitor the RF spectrum so as to manage the network for better spectral efficiency under the environment with various RF signals.

7.3 ZigBee Decoding

In this subsection, we validate the ability to decode ZigBee signals by SDR-Lite. We evaluate symbol error rate (SER) and packet reception rate (PRR). For detailed analysis, we use USRP B210 to record emulated header and ZigBee signal and SDR-Lite is used to reconstruct and decode the ZigBee signal.

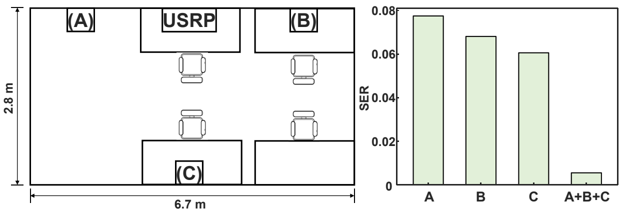
SER and PRR. For in-depth analysis and to rule out any irrelevant factors, we transmit the recorded emulated header and ZigBee signal using USRP, where various TX powers were tested. We set the distance between the USRP and SDR-Lite to 2 m. Figure 27(a) shows the SER performance with varied TX powers of the USRP from -5 dBm to 10 dBm. We note that as described in Section 7.2 the power of emulated header is required to be controlled considering the power of the target signal, and thus we set the power ratio between the emulated header and the ZigBee signal to 16:1. With 10 dBm TX power, we achieve the SER performance up to 95%, while the SER



(a) SER Plot (b) PRR Plot

Figure 27: ZigBee Decoding.

performance becomes 80% for the TX power of -5 dBm. Based on the SER performance, Figure 27(b) presents the PRR performance with different frame lengths for the TX power of 10 dBm. For a frame with 4 bytes, the PRR performance is 64%, while the PRR drops as the frame length gets longer. Three retransmissions offers PRR of above 99%, enabling reliable ZigBee decoding. We note that retransmissions are commonly used in ZigBee networks (e.g., to turn on duty-cycled receivers). This result demonstrates the ZigBee decoding capability of SDR-Lite, essentially transforming commodity WiFi device into a ZigBee decoder (e.g., mobile IoT reader).



(a) Multiple SDR-Lite Scenario **(b) Symbol Error Rate**

Figure 28: ZigBee Decoding with multiple SDR-Lite.

Reliable decoding with multiple SDR-Lites. We demonstrate vastly improving ZigBee decoding performance with multiple SDR-Lites, leveraging various WiFi devices in proximity. Specifically, multiple SDR-Lite are triggered by broadcasting emulated header, where they offer receiver diversity to significantly enhance the SER. Figure 28(a) presents the evaluation scenario. Following the previous experimental settings, we use USRP to send the emulated header followed by ZigBee signal, under the same power. We explore the channel diversity among A, B, and C by decoding with the majority voting (denoted as A+B+C in Figure 28(b)). By doing so, Figure 28(b) shows a greatly improved SER of 0.56 %, compared to the cases when the decoding is separately performed on each receiver (i.e., 7.75 %, 6.81 %, and 6.06 % for A, B, and C, respectively). To summarize, leveraging multiple SDR-Lite enables highly reliable ZigBee decoding.

7.4 RF Fingerprinting: Device Identification

Next, we demonstrate a device identification application using RF signatures captured by SDR-Lite and its performance. Here, we use the frame transient feature [1][61]. The frame transient is observed at the start and the end of each frame, which is determined by the hardware manufacturing process and imperfections. This feature can be unique even among devices with the same model. As a result, such frame transient feature can be used to fingerprint various WiFi devices including mobile phones and WiFi APs. Table 1 summarizes all WiFi devices used in our experiment. To compare the performance of SDR-Lite and USRP in device fingerprinting, we set up both SDR-Lite and USRP to capture the frame transient features of WiFi signals from these devices (See Figure 29).

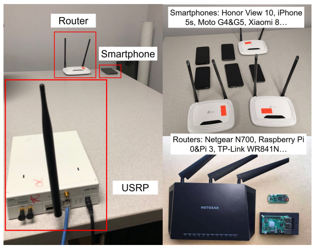
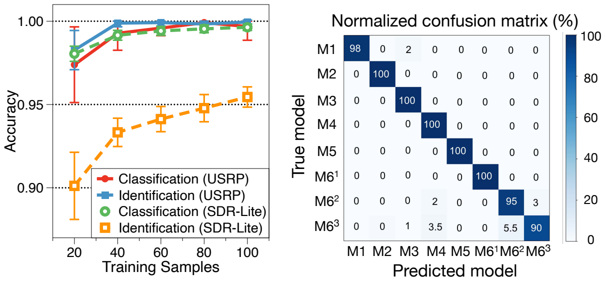

Router
Smartphone
USRP
Smartphones: Honor View 10, iPhone 5s, Moto G4&G5, Xiaomi 8…
Routers: Netgear N700, Raspberry Pi 0&Pi 3, TP-Link WR841N…

Figure 29: Device identification experiment.

Table 1: WiFi devices

Model	Quantity	Type
Honor View 10 (M1)	1	Smart device
iPad Pro (M2)	1	Smart device
iPhone 5s (M3)	1	Smart device
Moto G4 (M4)	1	Smart device
Moto G5 (M5)	1	Smart device
Xiaomi 8 (M6)	3	Smart device
TP-Link Archer A7 (A1)	1	Router
Netgear R7000 (A2)	1	Router
Raspberry Pi 0 (A3)	1	Router
Raspberry Pi 3 (A4)	1	Router
TP-Link WR841N (A5)	4	Router



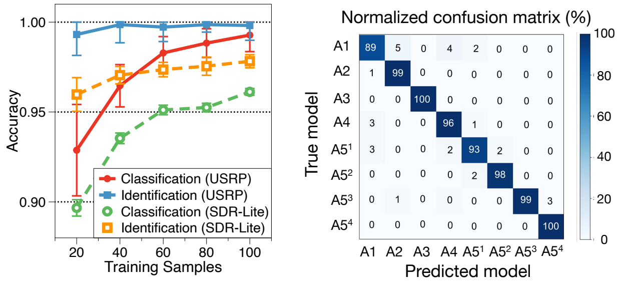
	M1	M2	M3	M4	M5	$M6^1$	$M6^2$	$M6^3$
M1	98	0	2	0	0	0	0	0
M2	0	100	0	0	0	0	0	0
M3	0	0	100	0	0	0	0	0
M4	0	0	0	100	0	0	0	0
M5	0	0	0	0	100	0	0	0
$M6^1$	0	0	0	0	0	100	0	0
$M6^2$	0	0	0	2	0	0	95	3
$M6^3$	0	0	1	3.5	0	0	5.5	90

(a) Accuracy (b) Confusion matrix

Figure 30: Mobile phone fingerprinting.

Device Identification. The device identification includes training process and testing process. We use transient signals captured by USRP and SDR-Lite as training sets, in which each transient signal from USRP and SDR-Lite has a length of 200 and 64 data points, respectively. Then, we capture another 200 samples from each device as a testing set and use support vector machine (SVM) algorithm to predict the device label of these transient signals. In the mobile phone fingerprinting experiment, we aim to classify 6 smart devices of different models (denoted by M1-M6), as well as 3 Xiaomi 8 phones ($M6^1$-$M6^3$), where the result is shown in Figure 30. In the router fingerprinting experiment, we classify 5 routers with different models (A1-A5), as well as 4 TP-Link WR841N routers ($A5^1$-$A5^4$). See Figure 31 for the result. The classification accuracy

[1] The portion of the time-domain signal when the transmitted signal envelop changes from one stable energy level to another stable energy level.



(a) Accuracy

(b) Confusion matrix

Figure 31: Router fingerprinting.

and identification accuracy denote the accuracy in classifying different models of devices and identifying individual devices with the same model, respectively. The results show that the accuracy increases with the growing number of training samples. SDR-Lite achieves very similar device classification accuracy as USRP, given sufficient training samples. However, for the identification in the same model, SDR-Lite shows the lower accuracy compared to that of USRP, while this degradation is caused by the inevitable signal distortion during the reconstruction process. It is noteworthy that, with 100 training frames, the accuracy of device identification for both smartphones and routers remarkably exceeds 94%.

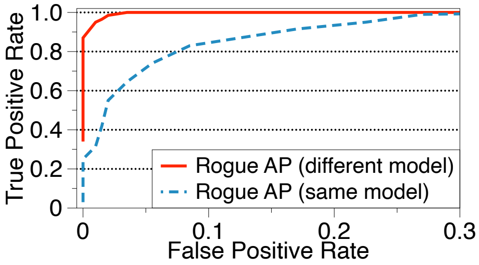


Figure 32: ROC curves for rogue AP detection.

Rogue AP Detection. Moreover, frame transients can be used to detect rogue APs. In this experiment, we let two TP-Link WR841N routers as legitimate APs and capture 100 transient samples through SDR-Lite as a training set for each device. Then, we choose another TP-Link WR841N router and a Raspberry Pi Zero as two rogue APs and use k-nearest-neighbor (kNN) algorithm to detect the rogue APs. The receiver operating characteristic (ROC) curves are presented in Figure 32, which shows that SDR-Lite achieves high detection accuracy when the rogue AP's model is different from legitimate APs. Specifically, SDR-Lite can detect the Raspberry Pi rogue AP with 99% true positive rate and less than 3% false positive rate. However, when the rogue AP's model is the same as the legitimate AP (i.e., WR841N), the detection performance degrades.

7.5 RF Fingerprinting: Drone Detection

Here we demonstrate another application of SDR-Lite as a drone detector. In general, commercial drones and their controllers use ISM band for message exchange (e.g., control, video streaming). In this evaluation, we choose three representative drone models: Intel Aero [31], DJI Mavic Pro [16], and 3DR Solo [1], where Intel Aero and DJI Mavic Pro use their proprietary protocols, while 3DR Solo adopts the WiFi standard for communication. Different from Section 7.4, in this application we introduce a DNN (Deep Neural Network) to perform RF fingerprinting.

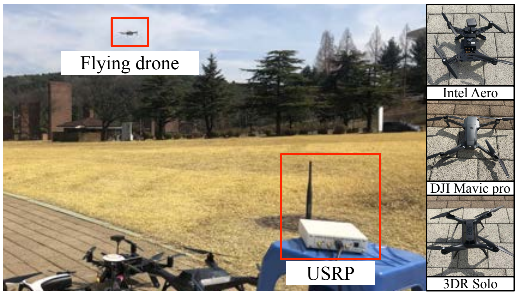
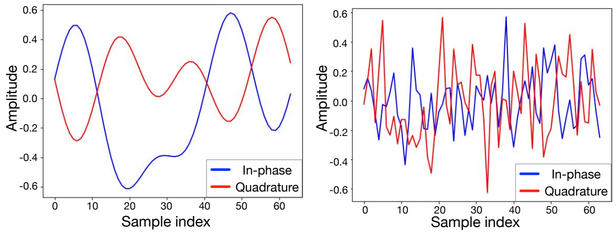


Figure 33: Drone detection experiment scenario.

Experiment Setup. Figure 33 shows our drone experimental scenario with the pictures of the three drones. In order to analyze time series RF signals, we choose 1D CNN (Convolutional Neural Network) [38] architecture. In this evaluation, we demonstrate two tasks of *(i) drone existence detection* and *(ii) drone model classification.* In the drone existence detection task, we scan the 2.4 GHz spectrum so as to detect a drone based on its RF signal. Drone model classification is to classify the model of drones by examining drones' RF signals. We use both USRP and SDR-Lite to scan the 2.4 GHz spectrum and capture the RF signal as an input of the DNN. Our evaluation compares the accuracy performances when drone signal is captured by USRP or SDR-Lite in the two tasks.



(a) Reconstructed Mavic Pro

(b) Reconstructed Solo

Figure 34: Reconstructed drone's signal by SDR-Lite.

Figure 34 demonstrates the reconstructed drone signals by SDR-Lite. As illustrated, the distinct characteristics are maintained in the reconstructed signals, since different drone models (e.g., Mavic Pro and Solo) normally adopt different PHY standards or modulations.

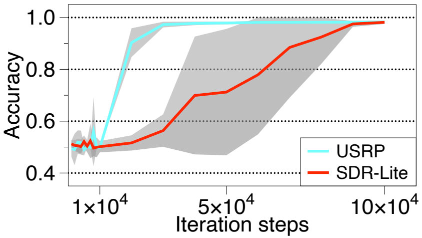
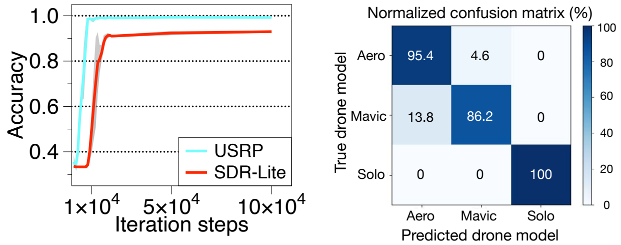


Figure 35: Drone detection accuracy.

Drone Existence Detection. To train a DNN, we collect 400 RF signal samples from Intel Aero and 3DR Solo drones, and 400 noise samples where each sample is composed of 64 In-phase and Quadrature points (as shown in Figure 34). The amplitude of every RF and noise signal is normalized, so that different RF gains of each signal does not impact the DNN outcomes. After training, we further collect 300 test samples to evaluate the accuracy performance for drone existence detection. To validate generality and avoid overfitting, the 100 test samples are collected from another Intel Aero drone which is different from the drone used for training sample collection, and another 100 test samples are generated from a DJI Mavic pro drone (not used in the training).

Figure 35 compares the accuracy of drone detection when the RF signal is captured by USRP and SDR-Lite (the shaded areas in grey represent standard deviation with 5 random seeds). With the increasing number of training iteration, the accuracy of USRP quickly converges to the performance of 98.2%. SDR-Lite achieves the similar performance (98%) compared to that of USRP while SDR-Lite requires more steps until convergence. This experiment demonstrates that SDR-Lite is able to detect drones' signal in the face of device heterogeneities.



(a) Drone classification (b) Confusion matrix

Figure 36: Model classification evaluation.

Drone Model Classification. In this experiment, we classify the model of the three drones by analyzing their RF signals through the DNN. Similar to the drone existence detection experiment, we collect 450 samples for training and 200 samples for test from each drone, respectively.

Figure 36(b) demonstrates the confusion matrix of the drone classification using the reconstructed signal by SDR-Lite where Y axis is the true drone model while X axis is the predicted model by the DNN. It is worth noting that the Solo drone can be classified without any error since it only uses WiFi signal which is significantly different from the other RF signals (as shown earlier). Figure 36(a) shows the performance evaluation of SDR-Lite and USRP as the number of iteration steps increases. While the USRP performance achieves the accuracy of 99.3%, SDR-Lite is able to classify the drone models with the accuracy of 93.7%. In comparison to the WiFi signal of Solo drone, it is more challenging to distinguish the RF signals of Aero and Mavic Pro drones, which bear more similarity especially when the signal is reconstructed by SDR-Lite. Even though the classification performance achieved by SDR-Lite is 93.7%, this classification is conducted by giving a 4 μs sample to the DNN; thus, we can further improve the accuracy by collecting multiple samples.

8 RELATED WORK

SDR Applications. There exist a number of commercially available SDR platforms [42, 53, 54], which require dedicated hardware. Recent development includes GalioT [43], Tick [65], IoT SDR [10], TinySDR [28], and Shadow WiFi [55]. Although relatively low-cost, TinySDR also requires a specially designed hardware, which only supports limited bandwidth up to 2 MHz. Shadow WiFi [55] enabled SDR transmitter functionalities on commodity WiFi, as opposed to SDR receiver functionalities offered in SDR-Lite. SDR has been used to collect RF signatures for device fingerprinting [7, 26, 36, 52, 61]. Interestingly, recent research uses SDR to detect drones by their RF signatures [48–50]. To summarize, SDR-Lite utilizes widely-accessible commodity WiFi to enable SDR receiver applications.

Cross-Technology Communication (CTC). Along with a plethora of heterogeneous wireless devices, cross-technology communication (CTC) has been spotlighted as a key technique to manage the network with high heterogeneity [8, 9, 18–20, 23–25, 29, 30, 33, 34, 37, 39, 40, 62, 69–72, 74, 75]. Among these studies, WEBee [39] proposes a signal emulation technique that transforms WiFi's OFDM signal into ZigBee signal. Inspired by it, SDR-Lite introduces the first OFDM emulation technique (i.e., WiFi's 40 MHz OFDM → WiFi's 20 MHz OFDM). More significantly, our signal emulation is not only be confined to CTC, but also can provide the SDR-like functionality using commodity WiFi devices.

Applications Using WiFi Channel State Information (CSI). CSI measures the channel between the transmitter and receiver pairs for equalization in coherent receivers, such as WiFi. Since the work on extracting CSI from commodity WiFi NICs [27, 66] as announced, there has been a body of recent studies exploring CSI to enable diverse application across various domain. This includes device or user identification [11, 12, 63, 73, 76] and localization [21, 35, 56, 67, 68]. Compared to CSI, SDR-Lite delivers orders of magnitude higher sampling rate. Specifically, a single CSI is obtained per packet (~325 Ksps), while SDR-Lite yields 17 Msps. This rich information enables SDR-Lite to support wide range of SDR receiver applications, well beyond of what CSI can offer. This includes (ZigBee) decoding and spectrum analysis an experimentally demonstrated in this paper.

9 CONCLUSION

We proposed SDR-Lite, a novel zero-cost and software-only SDR receiver built on commodity WiFi devices. For a WiFi device to capture ambient wireless signals, we design the new OFDM emulation technique which generates an emulated WiFi header within a WiFi packet payload. Upon receiving the emulated header, the WiFi receiver decodes the ambient signal in the air and outputs corresponding bit sequence. SDR-Lite reconstructs the I/Q of the ambient signal from the output bits, imitating a SDR receiver. The efficacy of SDR-Lite was demonstrated with seven SDR receiver applications under three categories including RF fingerprinting, spectrum monitoring, and ZigBee decoding.

ACKNOWLEDGMENTS

We thank our anonymous shepherd and reviewers for their constructive comments. This work was supported in part by the NSF under grants CNS-1717059, CNS-1950171, CNS-1949753, by the National Research Foundation of Korea (NRF) under grant (NRF-2020R1F1A1074657), and by Institute of Information & communications Technology Planning & Evaluation (IITP) under grant (No.2018-0-00772, Development of an ultra-Low power (500uW) long range radio for ultra-small IoT based on new Two-Tone OOK modulation scheme), and (IITP-2020-0-01787, the ITRC (Information Technology Research Center) support program).

REFERENCES

[1] 3DR. Solo drone. https://www.3dr.com/company/about-3dr/solo/.

[2] ALFA Network Inc. AWUS036NHA. http://www.rokland.com/support/specs/.

[3] Atheros. Ar 9271 datasheet. https://https://www.ath-drivers.eu/qualcomm-atheros-datasheets-for-AR9271.html.

[4] Atheros. Ar 9380 datasheet. https://www.ath-drivers.eu/download-driver-nr-309-for-atheros-AR9380-and-Windows10.html.

[5] M. M. Baig, H. GholamHosseini, and M. J. Connolly. Mobile healthcare applications: system design review, critical issues and challenges. *Australasian physical & engineering sciences in medicine*, 38(1):23–38, 2015.

[6] O. Barak and A. Touboul. Mobile broadband wireless network with interference mitigation mechanism to minimize interference within a cluster during multiple concurrent transmissions, Aug. 11 2009. US Patent 7,574,179.

[7] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 116–127, 2008.

[8] K. Chebrolu and A. Dhekne. Esense: communication through energy sensing. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 85–96. ACM, 2009.

[9] Y. Chen, Z. Li, and T. He. Twinbee: Reliable physical-layer cross-technology communication with symbol-level coding. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 153–161. IEEE, 2018.

[10] Y. Chen, S. Lu, H.-S. Kim, D. Blaauw, R. G. Dreslinski, and T. Mudge. A low power software-defined-radio baseband processor for the internet of things. In *2016 IEEE international symposium on high performance computer architecture (HPCA)*, pages 40–51. IEEE, 2016.

[11] L. Cheng and J. Wang. Walls have no ears: A non-intrusive wifi-based user identification system for mobile devices. *IEEE/ACM Transactions on Networking*, 27(1):245–257, 2019.

[12] J.-S. Choi, W.-H. Lee, J.-H. Lee, J.-H. Lee, and S.-C. Kim. Deep learning based nlos identification with commodity wlan devices. *IEEE Transactions on Vehicular Technology*, 67(4):3295–3303, 2017.

[13] CNN. Is it OK to shoot down a drone over your backyard? https://edition.cnn.com/2015/09/09/opinions/schneier-shoot-down-drones.

[14] D-Link. Dw-192 datasheet. https://support.dlink.com/ProductInfo.aspx?m=DWA-192.

[15] G. Diraco, A. Leone, and P. Siciliano. In-home hierarchical posture classification with a time-of-flight 3d sensor. *Gait & posture*, 39(1):182–187, 2014.

[16] DJI. Mavic Pro Drone. https://www.dji.com/mavic/info#specs.

[17] EDiMAX. Edimax ac-1200 datasheet. https://www.edimax.com/edimax/mw/cufiles/files/download/datasheet/EW-7822ULC_Datasheet_English.pdf.

[18] J. Elias, S. Paris, and M. Krunz. Cross-technology interference mitigation in body area networks: An optimization approach. *IEEE Transactions on Vehicular Technology*, 64(9):4144–4157, 2015.

[19] P. Gawlowicz, A. Zubow, and A. Wolisz. Enabling cross-technology communication between lte unlicensed and wifi. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 144–152. IEEE, 2018.

[20] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11 n robust to cross-technology interference. In *Proceedings of the ACM SIGCOMM 2011 conference*, pages 170–181, 2011.

[21] W. Gong and J. Liu. Roarray: Towards more robust indoor localization using sparse recovery with commodity wifi. *IEEE Transactions on Mobile Computing*, 18(6):1380–1392, 2019.

[22] G. W. Grube and T. W. Markison. Detection of unauthorized use of software applications in communication units, Jan. 3 1995. US Patent 5,379,343.

[23] 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.

[24] 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.

[25] X. Guo, X. Zheng, and Y. He. Wizig: Cross-technology energy communication over a noisy channel. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[26] J. Hall. *Detection of rogue devices in wireless networks.* PhD thesis, Carleton University, 2006.

[27] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR*, 41(1):53, 2011.

[28] M. Hessar, A. Najafi, V. Iyer, and S. Gollakota. Tinysdr: Low-power SDR platform for over-the-air programmable iot testbeds. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 1031–1046, 2020.

[29] A. Hithnawi, S. Li, H. Shafagh, J. Gross, and S. Duquennoy. Crosszig: Combating cross-technology interference in low-power wireless networks. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12. IEEE, 2016.

[30] Y. Hou, M. Li, X. Yuan, Y. T. Hou, and W. Lou. Cooperative cross-technology interference mitigation for heterogeneous multi-hop networks. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 880–888. IEEE, 2014.

[31] Intel. Aero drone. https://www.intel.com/content/www/us/en/support/products/97174/drones/development-drones/intel-aero-products.html.

[32] 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.

[33] W. Jiang, Z. Yin, S. M. Kim, and T. He. Transparent cross-technology communication over data traffic. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[34] 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 ACM SenSys*, 2017.

[35] C. R. Karanam, B. Korany, and Y. Mostofi. Magnitude-based angle-of-arrival estimation, localization, and target tracking. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 254–265, 2018.

[36] K. Kim, C. M. Spooner, I. Akbar, and J. H. Reed. Specific emitter identification for cognitive radio with application to ieee 802.11. In *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, pages 1–5. IEEE, 2008.

[37] 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.

[38] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman. 1D convolutional neural networks and applications: A survey. *arXiv preprint arXiv:1905.03554*, 2019.

[39] Z. Li and T. He. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of ACM MobiCom*, 2017.

[40] Z. Li and T. He. Longbee: Enabling long-range cross-technology communication. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 162–170. IEEE, 2018.

[41] E. Magistretti, K. K. Chintalapudi, B. Radunovic, and R. Ramjee. Wifi-nano: reclaiming wifi efficiency through 800 ns slots. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 37–48, 2011.

[42] Myriad-RF. LimeSDR. https://wiki.myriadrf.org/LimeSDR-Mini.

[43] R. Narayanan and S. Kumar. Revisiting software defined radios in the iot era. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pages 43–49, 2018.

[44] NESDR. Nesdr mini 2 +. https://www.nooelec.com/store/sdr/sdr-receivers/nesdr-mini-2.html.

[45] NESDR. Nesdr nano 2. https://www.nooelec.com/store/nesdr-nano2.html.

[46] NESDR. Nesdr nano 3. https://www.nooelec.com/store/nesdr-nano-three.html.

[47] F. News. Hidden cameras: Are you being watched? https://www.foxnews.com/tech/hidden-cameras-are-you-being-watched.

[48] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu. Investigating cost-effective rf-based detection of drones. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 17–22, 2016.

[49] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu. Matthan: Drone presence detection by identifying physical signatures in the drone's rf communication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 211–224, 2017.

[50] M. Peacock and M. N. Johnstone. Towards detection and control of civilian unmanned aerial vehicles. In *The Proceedings of 14 th Australian Information Warfare Conference*, page 9. Citeseer, 2013.

[51] D. Pfammatter, D. Giustiniano, and V. Lenders. A software-defined sensor architecture for large-scale wideband spectrum monitoring. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 71–82, 2015.

[52] K. B. Rasmussen and S. Capkun. Implications of radio fingerprinting on the security of sensor networks. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*, pages 331–340. IEEE, 2007.

[53] E. Research. USRP N210. https://www.ettus.com/product/details/UN210-KIT, n.d.

[54] A. Sabharwal. WARP 802.11 Physical Layer. http://warpproject.org/trac/wiki/802.11/PHY, 2016.

[55] M. Schulz, J. Link, F. Gringoli, and M. Hollick. Shadow wi-fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over wi-fi. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, page 256–268, 2018.

[56] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, page 376–388, 2018.

[57] M. Sruthi, M. Abirami, A. Manikkoth, R. Gandhiraj, and K. Soman. Low cost digital transceiver design for software defined radio using rtl-sdr. In *2013 international mutli-conference on automation, computing, communication, control and*

compressed sensing (iMac4s), pages 852–855. IEEE, 2013.

[58] Texas Instruments. SimpleLink multi-standard CC2650 SensorTag. http://www.ti.com/tool/TIDC-CC2650STK-SENSORTAG.

[59] T. N. Y. Times. Two Major Saudi Oil Installations Hit by Drone Strike, and U.S. Balems Iran. https://www.nytimes.com/2019/09/14/world /middleeast/saudi-arabia-refineries-drone-attack.html.

[60] T. N. Y. Times. White House Drone Crash Described as a U.S. Worker's Drunken Lark. https://www.nytimes.com/2015/01/28/us/white-house-drone.html.

[61] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir. Fingerprinting wi-fi devices using software defined radios. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 3–14, 2016.

[62] 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.

[63] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu. E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, page 617–628, 2014.

[64] Wireless LAN Working Group. Ieee standard part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[65] H. Wu, T. Wang, Z. Yuan, C. Peng, Z. Li, Z. Tan, B. Ding, X. Li, Y. Li, J. Liu, et al. The tick programmable low-latency sdr system. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 101–113, 2017.

[66] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wifi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, page 53–64, New York, NY, USA, 2015. ACM.

[67] Y. Xie, J. Xiong, M. Li, and K. Jamieson. Md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, 2019.

[68] Y. Xie, J. Xiong, M. Li, and K. Jamieson. md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[69] P. Yang, Y. Yan, X.-Y. Li, Y. Zhang, Y. Tao, and L. You. Taming cross-technology interference for wi-fi and zigbee coexistence networks. *IEEE Transactions on Mobile Computing*, 15(4):1009–1021, 2016.

[70] Z. Yin, W. Jiang, S. M. Kim, and T. He. C-morse: Cross-technology communication with transparent morse coding. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[71] Z. Yin, Z. Li, S. M. Kim, and T. He. Explicit channel coordination via cross-technology communication. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 178–190, 2018.

[72] Y. Yubo, Y. Panlong, L. Xiangyang, T. Yue, Z. Lan, and Y. Lizhao. Zimo: building cross-technology mimo to harmonize zigbee smog with wifi flash without intervention. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 465–476. ACM, 2013.

[73] Y. Zeng, P. H. Pathak, and P. Mohapatra. Wiwho: Wifi-based person identification in smart spaces. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–12, 2016.

[74] Y. Zhang and Q. Li. Howies: A holistic approach to zigbee assisted wifi energy savings in mobile devices. In *INFOCOM, 2013 Proceedings IEEE*, pages 1366–1374. IEEE, 2013.

[75] 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.

[76] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. J. Spanos. Wifi-based human identification via convex tensor shapelet learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.