Exploiting Frame Preamble Waveforms to Support New Physical-layer Functions in OFDM-based 802.11 Systems

Hanif Rahbari and Marwan Krunz, *Fellow, IEEE*

***Abstract*—The frame preamble in current WiFi systems is designed to facilitate various PHY-layer functions, including frequency offset estimation and frame detection. However, this preamble is typically fixed and is never used to convey any user-specific bits. Embedding information into the preamble opens the door for several new PHY-layer applications. For example, the PHY header no longer needs to be transmitted at a known (lowest) rate if this rate can be announced earlier in the preamble. A full-duplex transmitter can use the embedded information to inform other devices of its current operation mode (e.g., transmit/receive vs. transmit/sense), obviating the need for additional control packets. In security applications, a PHY-layer sender identifier can be embedded in the preamble to facilitate PHY-level encryption.**

However, modifying the standard preamble to embed user information may disrupt the operation of 802.11a/n/ac devices. In this paper, we propose *P-modulation*, a method that enables an OFDM-based 802.11 transmitter to embed up to 19 user-specific bits in the frame preamble while maintaining the highest reliability required by the system. The proposed *P-modulation* is backward-compatible with legacy receivers. Our analysis and USRP-based experimental results confirm the practicality of the scheme. Our scheme also provides insights into designing time-varying preambles for future wireless systems.

I. Introduction

WIRELESS systems continue to boost their performance and enhance their security at the physical (PHY) layer by employing various techniques, including MIMO and artificial noise for improved transmission security. However, in these wireless systems, the preamble is never utilized for conveying user-generated bits at the PHY layer; a feature that can facilitate new functionalities related to enhanced performance and/or security. In a wireless system, the frame preamble is a special and often publicly known signal, prepended to the (modulated) PHY-layer header at the transmitter (Tx) and used by the receiver (Rx) to perform several PHY-layer functions. These functions include frame detection, frequency offset (FO) estimation, channel state information (CSI) estimation, dynamic range estimation (used for automatic gain control (AGC) convergence), etc. In this paper, we consider the frame preamble in OFDM-based WiFi systems (802.11a/n/ac). The duration of this preamble in 802.11a [1] is 16 μs and can be up to 52 μs in 802.11n/ac MIMO systems [2], [3], irrespective of the size and transmission rate of the PHY-layer payload. So depending on the payload size and the transmission rate, the preamble in 802.11a systems can take up to 50% of the frame duration, as shown in Table I, and even more in 802.11n/ac MIMO systems. Because in current systems the frame preamble is assumed to be constant and is not designed to carry any user-generated bits, it is under-utilized.

TABLE I
Percentage of the preamble duration in a frame for different PHY-layer payload sizes and data rates (802.11a systems).

Packet type	Size (bytes)	6 Mbps	18 Mbps	36 Mbps	54 Mbps
TCP ACK	72	13.8%	30.8%	44.4%	50%
WLAN MTU	2324	0.5%	1.5%	3.0%	4.3%

The preamble of relatively obsolete non-OFDM (single-carrier) 802.11b systems was considered in [4] for embedding a user-defined bit sequence. This preamble is based on Barker sequences and must be fully known to the Rx for estimating the CSI, which implies that the scheme in [4] is not backward-compatible with 802.11b devices. In contrast, in here we focus on the preamble of widespread OFDM-based IEEE 802.11 systems (e.g., .11n/ac), whose structure is fundamentally different from the one used in 802.11b systems, and we study the feasibility of using this preamble to carry a sequence of user-defined bits. For the first time, we exploit the potential of this preamble in modulating a bit sequence while preserving its basic properties and backward-compatibility (legacy/unaware receivers still operate as normal). We further suggest a set of new PHY-layer functions that can be enabled by employing our proposed embedding technique. Our scheme is also different from Frequency Offset Embedding for Authenticating Transmitters [5] in that the Rx in our scheme does not require receiving multiple frames or even one full frame to estimate the true FO and then extract the embedded bits. Moreover, our scheme achieves a higher per-frame embedding capacity than the 1–4 bits in [5] and more robustness against time-selective fading channels.

In OFDM-based 802.11 systems, the Rx needs to know the preamble "structure" but does not need to know the exact signal of the whole preamble. We exploit this feature to construct several new but compliant preamble waveforms. Each waveform can represent the modulated version of a bit sequence. To generate these waveforms, we design a novel modulation technique called *preamble modulation* (*P-modu-*

H. Rahbari is with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA e-mail: rahbari@vt.edu.

M. Krunz is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, USA e-mail: krunz@email.arizona.edu.

This research was supported in part by the National Science Foundation (grants # CNS-1409172 and IIP-1265960) and Raytheon. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

lation) that is applicable to the common preamble structure of 802.11a/n/ac systems. *P-modulation* is a combination of two independent signal processing techniques, time shift in the time domain and phase rotation in the frequency domain. These techniques preserve the key characteristics of a standard preamble. At the Rx and after applying channel equalization to the received preamble signal, *P-modulation* exploits the particular dependency pattern among the subcarriers of preamble waveforms (the intrinsic redundancy) and also the mandatory repetitions in the preamble to efficiently distinguish between different legitimate waveforms. This way, it can demodulate the embedded information bits. Furthermore, *P-modulation* employs a simple low-complexity fine-scale synchronization technique to account for the sensitivity of its demodulator to timing errors. For the same level of robustness to channel/device impairments as BPSK modulation, *P-modulation* can embed up to 8 and 19 bits in the preamble when the channel bandwidth is 20 MHz and 80 MHz (e.g., 802.11ac), respectively. So the frame loss rate under *P-modulation* will not be worse than the loss rate of a system with BPSK-modulated header. More user bits can be embedded if *P-modulation* is to be designed to achieve the same robustness of higher-order modulation schemes.

The rest of the paper is organized as follows. In Section II, we motivate the significance of *P-modulation* by discussing various potential applications. In Section III, we provide background on preamble functions and the requirements that must be met in a compliant OFDM-based WiFi preamble. We then introduce *P-modulation* and its related modulation/demodulation techniques in Section IV. The robustness of this scheme to channel/device impairments and its extension to 802.11 MIMO systems are discussed in Sections V and VI, respectively. Finally, we present the results of extensive simulations and USRP experiments in Section VII. The paper is concluded in Section VIII.

II. New PHY-layer Applications of P-modulation

Embedding/modulating user-specific information in the frame preamble paves the way for several important applications. We divide these applications into two categories: Security and PHY-layer (pre-header) signaling.

1) *Security*– One of the prominent security applications of *P-modulation* is that the embedded bit sequence can represent a (time-varying) PHY-layer sender identifier, hence facilitating PHY-level frame encryption [4] and preventing MAC spoofing attacks. Such encryption is impossible in the absence of a reliable PHY-layer identifier. The embedded bit sequence can also be used to represent other new parameters at the PHY layer, such as:

- Sender's time-varying digital signature (used to authenticate a link and prevent copycat/replay attacks [6])
- Initialization vector for assigning the locations of secret pilot subcarriers or enabling cryptographic interleaving to mitigate pilot tone [7], [8] and interleaving [9] jamming attacks, respectively, in OFDM systems
- Seed for pseudo-random number generators, e.g., to adapt the frequency-hopping pattern over a secure link, to facilitate untraceable convolutional coding and modulation obfuscation [4], to enable crypto-based PHY-layer techniques such as [10], etc.

In addition, time-varying preamble waveforms that change from one frame to another can mitigate FO estimation attacks that rely on publicly known preamble signal to craft a jamming signal and efficiently disrupt the FO estimation process in OFDM systems [11], [12].

As an example of these use cases, we now discuss the need for having a reliable sender identifier at the PHY layer. Sender identification is a key functionality in any wireless network. It allows the Rx to distinguish between different transmitting nodes. If nodes were to employ full-frame encryption to prevent the leakage of side-channel information, sender identification is required at the Rx before the decryption can take place so as to look up the right decryption key. In this case, the Rx needs to receive a plaintext sender identifier at some header in the protocol stack before it can start the decryption process. For example, in IEEE WLAN standards, the globally unique MAC address acts as the sender identifier at the link layer. However, such an identifier is extracted only after decoding the PHY-layer header. Thus, if the frame is to be fully encrypted (including its PHY header), the MAC address cannot be used for identification. In other words, the PHY-layer header cannot be decrypted until the MAC address has been obtained, creating a deadlock situation. Hence, relying on a link-layer identifier necessitates transmitting the PHY-layer header in the clear. However, an adversary can fingerprint a device/user [13] using the unencrypted PHY-layer header and can apply traffic analysis to disclose several private information, even if the frame payload is encrypted [13]–[15]. Transmitting the PHY-layer header in the clear also makes the system vulnerable to various selective jamming attacks (e.g., [16]).

Furthermore, a plaintext and predictable MAC identifier opens the door for MAC spoofing and/or unauthorized user tracking attacks. The case of trash cans in London in 2013 is an example of such attacks [17]. The can suppliers had installed a sniffer in the cans to collect information from smartphones of people walking in London's Square Mall, mostly based on PHY-layer header fields. The intention was to study customers' shopping habits and generate targeted advertisements. The seriousness of these tracking attacks has been recently acknowledged by IEEE and IETF, and accordingly, a new study group was formed to assess the privacy implications of visible MAC addresses and other link-layer privacy issues [18]. To provide a more secure link-layer identification approach and prevent user tracking, this group suggested using random MAC addresses, generated based on a chain of unpredictable but unencrypted time-rolling identifiers (e.g., [19]–[22]). As discussed earlier, however, link-layer identification precludes the feasibility of full-frame encryption. In addition, implementation of MAC address randomization on commercial devices (e.g., for hiding the true address in probe requests and location scans in Apple iOS 9) have been shown to exhibit several vulnerabilities [23]. By employing *P-modulation*, those random (time-varying) identifiers can instead be used at the PHY layer, enabling full-frame encryption.

A number of PHY-layer sender identification methods (or authentication methods to defend against identity-based spoofing) have been proposed in the literature. For example, several channel-based sender authentication schemes (e.g., [24]–[26]) have been suggested. These schemes, however, are often impractical due to node mobility, correlated CSI (signature) in a vicinity of the Tx [27], and CSI estimation errors at the Rx. Hardware-based (radiometric) sender authentication schemes (e.g., [28]) exploit the inherent device-specific manufacturing impairments for authentication. However, the inaccuracy in the measurements of COTS radios prevents successful deployment of these methods [29]. In contrast, *P-modulation* offers a more reliable PHY-layer platform for sender identification (authentication), which is robust to node mobility and is independent of the CSI and Tx hardware.

2) *PHY-layer (pre-Header) Signaling– P-modulation* can be used as a signaling mechanism for certain PHY-layer operations, which otherwise require modifying existing header structures and introducing new fields. For example, the embedded bit sequence can be used to convey the operation mode of the Tx in full-duplex communications (e.g., transmit/receive vs. transmit/sense), the frame-specific pattern of *traveling pilots* proposed for the upcoming IEEE 802.11ah standard to improve channel estimation under high Doppler scenarios [30], or any PHY-layer field required for future applications that cannot be conveyed in the standard PHY-layer header. Alternatively, the embedded bits can be a part of the PHY header, merged into the preamble and removed from the frame (to reduce the frame duration). Such an approach increases the utility of the frame by communicating a part of it through the preamble. In addition, *P-modulation* can enable the Tx to use a higher transmission rate for the PHY-layer header, which is transmitted at a known (often the lowest) rate in existing systems, thus reducing the frame duration by announcing this frame-specific rate in the preamble.

III. Preliminaries

Before introducing *P-modulation*, we first explain the key operations and special characteristics based on which the preamble of an OFDM-based IEEE 802.11 frame is designed. The preamble consists of two fields [1], [2]: a *short training field* (STF), which consists of 10 repetitions of some periodic *short training signal* (STS), and a *long training field* (LTF), which includes two repetitions of another periodic training signal (see Fig. 1 for the preamble of a 20 MHz channel). The signal for the STF is constructed by transmitting on only a quarter of the subcarriers (one every four subcarriers), resulting in a larger greatest common divisor (gcd) of the subcarrier frequencies than the gcd of the LTF subcarrier frequencies, and hence a shorter time period. The preambles in 802.11n and 802.11ac (MIMO) standards are essentially the same as 802.11a but are transmitted over a wider bandwidth (up to 160 MHz). They may also include an additional STF for better AGC and multiple LTFs for channel sounding and backward compatibility. After performing the STF functions described below, LTS is used for CSI and fine FO estimation; hence, the LTF signal must be known in advance at the Rx. On the other hand, the STF does not need to be fully known to the Rx, as will be discussed next.

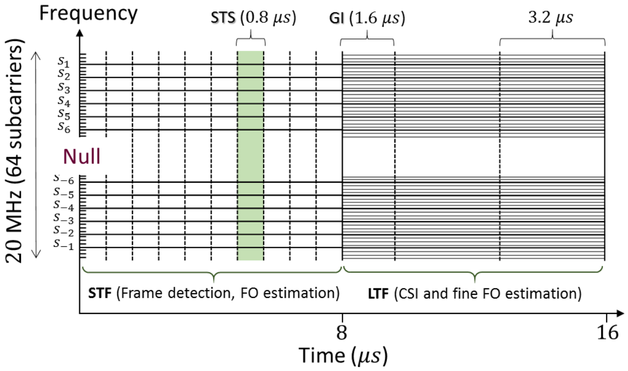


Fig. 1. Preamble structure and subcarriers in 802.11a. s_i refers to the ith subcarrier in the STF, $i \in \{-6, \ldots, 6\}/\{0\}$.

A. STF Functions

The STF is used for frame detection, coarse FO estimation, AGC, diversity selection, and other functions. Accurate frame detection and FO estimation are two key operations that require two identical signals in the STF (e.g., two STSs). Having identical signals enables the Rx to detect the frame without knowing the channel.

Let h represent the channel[1] between the Tx and the Rx, and let $x(t)$ and $n(t)$ be the transmitted signal and the noise, respectively, at time t. In autocorrelation-based frame detection, the Rx needs only to know the period, denoted by λ_S, of the STF signal to perform frame detection [31]. It searches for identical STSs by correlating samples of the received signal $h\,x(t)+n(t)$, separated by a time lag that equals to a multiple of λ_S. Let $\mathcal{A}_\tau(t)$ be the autocorrelation value at time t and time lag τ:

$$\mathcal{A}_\tau(t) = \sum_{m=0}^{\tau-1} \frac{|h|^2\, x^*(t+m)\, x(t+\tau+m) + n_\tau(t+m)}{\sum_{q=0}^{\tau-1} |h\, x(t+\tau+q) + n(t+\tau+q)|^2} \qquad (1)$$

where the time unit of the integer values λ_S, τ, and m is the sampling period at the Rx and $n_\tau(t) \triangleq h^* x^*(t)\, n(t+\tau) + n^*(t)\, h^* x^*(t+\tau) + n^*(t)\, n(t+\tau)$. The time t at which $\mathcal{A}_\tau$ peaks is taken as the start of the frame.[2] Assuming the channel does not change during the STF, this method is channel-independent. It is adopted in practical systems.

The other key function of the STF is FO estimation. FO often refers to the inherent difference in the operating frequencies of two oscillators. It creates a time-varying phase offset in the received signal, and can significantly harm the performance of an OFDM system [31]. To estimate the FO after determining the start of the STF, the Rx considers the samples of two consecutive STSs and measures how the phases of the samples of the first STS change after λ_S seconds. The average of the phase changes captures the devices' FO.

[1] In WLAN settings, the channel's coherence time is almost always larger than the duration of the STF (8 μs). A coherence time of 8 μs would be equivalent to a speed larger than 10^6 m/s, which is quite unlikely. So in this paper, we can practically assume that the channel is time-invariant, and use h instead of $h(t)$.

[2] To account for noise, in practical systems the first time that $\mathcal{A}_\tau$ exceeds a near-to-maximum threshold is taken as the frame start time. If this instance happens to be before the true start time, the Rx can use the LTF to correct it.

B. STF Requirements

The STF in current IEEE standards (as defined in [1]) is designed to satisfy certain requirements related to the preamble functions. Any modification in this design should take these requirements into account. In the following, we discuss these requirements and explain how the current STF design satisfies them.

1) Periodicity: In 802.11a, an OFDM symbol consists of 64 orthogonal frequency subcarriers (up to 256 subcarriers in 802.11n and 802.11ac). Without considering the guard interval, the symbol lasts for 3.2 μs. Twelve subcarriers are not used (the null subcarriers in Fig. 1). The period λ_S of an STS is set to $3.2/4 = 0.8\ \mu$s by using only 12 equally-spaced subcarriers out of 52 active ones. This enables the Rx to cover a wide range of FOs, up to 625 kHz [32]. In addition, autocorrelation-based frame detection and FO estimation rely on identical repetitions of the same STS. So all the ten STSs should be identical.

Let $\mathcal{S} = [s_{-6}, \ldots, s_{-1}, s_1, \ldots, s_6]$ be the symbol sequence carried in these 12 subcarriers. Let $P(t)$ be the value of the STF signal at time t. Then,

$$P(t) = \sum_{k=-6, k\neq 0}^{6} s_k e^{2\pi j(4k)\Delta_f t}, \text{ for } t \in [0, 8]\,\mu s \qquad (2)$$

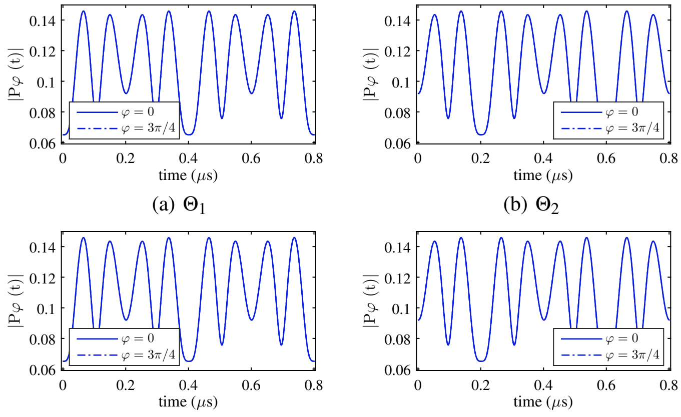
where Δ_f is the frequency spacing between any two successive subcarriers in the original 64 subcarriers.

2) Peak-to-average Power Ratio (PAPR): Due to the non-linearity of the power amplifier at the Tx, the PAPR of the STF, denoted by R_{PAP}, should be as small as possible to avoid poor transmission. IEEE 802.11a/g/n/ac standards use a set of 12 QPSK-modulated symbols for the sequence $\mathcal{S}$ (as shown in Table II). For 802.11a/g, this set results in $R_{\text{PAP}} = 2.24$ dB [32]. These symbols are multiplied by a factor of $\sqrt{13/6}$ to normalize the average power of the STF with respect to the rest of the frame [1].

3) Dynamic range: The STF is also used for AGC convergence. In order to accelerate AGC locking and adjusting the reference signal value for the A/D converter at the Rx, the whole dynamic range of the STF, denoted by R_{DR}, should be covered by the A/D converter resolution and without any overflow/underflow [32]. For 802.11a/g, the resulting dynamic range for the symbol sequence $\mathcal{S}$ (shown in Table II) is $R_{\text{DR}} = 7.01$ dB, one of the lowest possible dynamic ranges that can be achieved by candidate sequences $\mathcal{S}$ of low R_{PAP}. In HT-mixed format of 802.11n, an additional STF is used to improve AGC estimation in MIMO systems [2].

IV. Proposed Preamble-Modulation Scheme

We now introduce *P-modulation* and show how it modulates (embeds) a bit sequence in the STF of the preamble at the Tx and then demodulates (extracts) this sequence at the Rx with low complexity and very little impact on normal preamble functions (for backward compatibility with legacy receivers). *P-modulation* maintains all the STF requirements stated above, but modifies other properties of the STF.



(a) Θ_1 (b) Θ_2

(c) Θ_3 (d) Θ_4

Fig. 2. Amplitudes of the four STFs generated using the patterns in Table III. (Only one STS is shown.)

A. Sequence Modulation

In our design, modulating a bit sequence is different from conventional digital modulation, where one or more bits are mapped into a symbol, independent of other symbols. Instead, we use a particular "chain" of correlated symbols to modulate a bit sequence. The correlation among symbols in $\mathcal{S}$ is impacted by the R_{PAP} and R_{DR} requirements stated above. In the following, we say a signal is *STF-compliant* if it satisfies all the requirements in Section III. The number of distinct STF-compliant signals determines the number of different bit sequences that can be embedded in the STF.

To construct a set of STF-compliant sequences, we first find a set of compliant signals. For these signals, we identify the different dependency patterns among the symbols in $\mathcal{S}$. Using these patterns, we then employ two signal processing techniques to generate as many compliant signals as possible. In here, a dependency pattern is defined as the sequence of (wrapped) phase differences between the successive symbols in $\mathcal{S}$. Let θ_i represent the ith phase difference and let $\Theta = (\theta_1, \ldots, \theta_{11})$ represent a dependency pattern in $\mathcal{S}$, starting from s_{-6}. For example, $\theta_1 = \measuredangle(s_{-5}) - \measuredangle(s_{-6})$, where $\measuredangle(.)$ indicates the phase of a complex symbol (for illustration purposes, we henceforth use the IEEE 802.11a STF waveform). Therefore, a set $\mathcal{S}$ can be alternatively represented using its first symbol s_{-6} and Θ, as follows:

$$s_i = \begin{cases} e^{j\theta_{i+6}} s_{i-2}, & \text{for } i = 1 \\ e^{j\theta_{i+6}} s_{i-1}, & \text{for } i \in \{-5, -4, \ldots, 6\}/\{0, 1\} \end{cases} \qquad (3)$$

Through exhaustive search among all 4^{12} possible sequences that consist only of QPSK symbols[3], we identified 16 STF-compliant signals. In these signals, s_{-6} can be any of the four possible QPSK symbols with equal probability. We further recognize four distinct (but not independent) dependency patterns: $\Theta_1, \ldots, \Theta_4$, where each pattern is characterized by one of the possible values for θ_1 (see Table III). For example, the dependency pattern of the sequence in Table II is Θ_3. As will be discussed shortly, θ_i's also depend on θ_1.

[3] Considering symbols of higher-order modulation schemes would make the exhaustive search intractable. We instead identify other compliant signals through various signal processing techniques.

TABLE II

SEQUENCE OF QPSK-MODULATED SYMBOLS USED TO GENERATE STF IN 802.11A/G/N/AC [1]–[3]. $|s_{-6}| = |s_{-5}| = \ldots = |s_6| = \sqrt{2}$. THIS SEQUENCE IS THEN MULTIPLIED BY $\sqrt{13/6}$ TO NORMALIZE THE AVERAGE POWER OF THE RESULTING OFDM SYMBOLS.

s_i	s_{-6}	s_{-5}	s_{-4}	s_{-3}	s_{-2}	s_{-1}	s_1	s_2	s_3	s_4	s_5	s_6
Value	$1+j$	$-1-j$	$1+j$	$-1-j$	$-1-j$	$1+j$	$-1-j$	$-1-j$	$1+j$	$1+j$	$1+j$	$1+j$

TABLE III

DEPENDENCY PATTERNS AMONG ALL POSSIBLE COMBINATIONS OF QPSK-MODULATED SYMBOLS IN $\mathcal{S}$ THAT SATISFY THE 802.11A/G STF REQUIREMENTS. THE IEEE STANDARD USES THE DEPENDENCY PATTERN Θ_3 AND $\varphi = 0$ WHEN $s_{-6} = 1 + j$.

Θ_i	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}	θ_{11}	b_2b_1
$i = 1$	0	0	0	π	0	π	π	0	π	π	π	00
$i = 2$	$\pi/2$	$\pi/2$	$\pi/2$	$-\pi/2$	$\pi/2$	0	$-\pi/2$	$\pi/2$	$-\pi/2$	$-\pi/2$	$-\pi/2$	01
$i = 3$	π	π	π	0	π	π	0	π	0	0	0	11
$i = 4$	$-\pi/2$	$-\pi/2$	$-\pi/2$	$\pi/2$	$-\pi/2$	0	$\pi/2$	$-\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	10

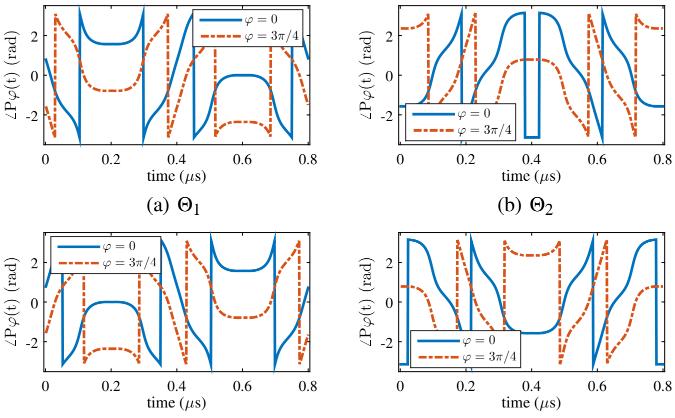
To design more compliant signals, we use the same dependency patterns $\Theta_1, \ldots, \Theta_4$ but exploit the fact that if the symbols transmitted on the subcarriers of an OFDM symbol are all phase-shifted by the *same* amount, then the period, R_{PAP}, and R_{DR} of that OFDM symbol do not change. Hence, we can use higher-order PSK symbols as s_{-6} (a somewhat similar approach was briefly discussed in [33] to establish covert channels in WiFi systems). To illustrate, let φ be the introduced phase shift of the elements in $\mathcal{S}$ and let $P_\varphi(t)$ be the new STF after this shift. Then,

$$P_\varphi(t) = e^{j\varphi} P(t). \tag{4}$$

Multiplying a signal by a constant coefficient does not change the ratio of the maximum and minimum amplitude of the signal (i.e., R_{DR}) or the ratio of the maximum and the root-mean-square of the signal (i.e., R_{PAP}). So the Tx can select any phase for s_{-6} and the same amplitude of $\sqrt{2}$, follow one of the patterns Θ_i, $i = 1, \ldots, 4$, to define the rest of the symbols in $\mathcal{S}$ and generate an STF-compliant signal.

In Fig. 2, we show the amplitudes of the STF-compliant signals, constructed by using two different values for φ and dependency patterns Θ_1 to Θ_4. These figures also show that $P_\varphi(t)$ and $P(t)$ with the same dependency pattern have the same envelope. Hence, amplitude-based STF functions (e.g., frame detection and FO estimation) will not be impacted by the phase shift. The specific selection of the pattern Θ_3 in the IEEE 802.11 standards is dictated by cross-correlation-based detection issues (e.g., matched filter performance in the boundary region between the STF and the LTF [34]). However, by using the autocorrelation method for frame detection at the Rx, those issues will not be binding for us.

The coefficient $e^{j\varphi}$ rotates the constellation map of the symbols in $\mathcal{S}$ by φ degrees. Therefore, the set of s_{-6} values that can be used to generate STF-compliant signals consists of the symbols of a PSK modulation scheme. The order of this modulation scheme, denoted by M, depends on the performance of the PSK demodulation operation and the accuracy of pattern detection (discussed in Section IV-B), as well as the accuracy of CSI and FO estimation (discussed in Section V). The order specifies how many bits can be modulated using different constellation rotations when using the same pattern. We refer to these $\log_2 M$ bits as *rotation bits*. The Rx can exploit the correlation among the symbols in $\mathcal{S}$ and use all of them to improve the demodulation accuracy.



(a) Θ_1

(b) Θ_2

(c) Θ_3

(d) Θ_4

Fig. 3. Phases of the four STFs generated using the patterns in Table III. (Only one STS is shown.)

However, incorrect detection of the underlying pattern may significantly change the expected sequence of phases. This can be seen in Fig. 3, where we plot the phases of the signals in Fig. 2.

To identify additional STF-compliant signals, we take advantage of the fact that the patterns in Table III are indeed not independent and a pattern Θ_i can be obtained by adding a multiple of $\pi/2$ to Θ_1 (for θ_6 a multiple of π should be added). For example, $\Theta_2 = \Theta_1 + \pi/2$. In other words, when the phase differences between successive symbols of the frequency subcarriers are changed by a common constant, the resulting signal remains STF-compliant. To explain, we note that adding a constant to Θ_i in the frequency domain has an interpretation in the time domain: In OFDM systems, a time shift in the signal results in a *linear* phase shift in the symbols along the ordered subcarriers [35], and vise versa. Because the elements in $\mathcal{S}$ are ordered according to their frequencies, such a linear shift brings about the same change in phase differences θ_i, $i = 1, \ldots, 11$. This amount of change is indeed the slope of a line that defines a symbol's phase shift with respect to its frequency. Now let N be the number of samples in a symbol and ν be the line slope. The corresponding time shift t_s (measured in the number of samples) is $t_s = \frac{N \times \nu}{2\pi}$. For example, the dependency patterns in Table III represent different amounts of time shift of the same signal, as shown in Fig. 2. Note that cyclic time shifts in a periodic signal do not change its dynamic range and PAPR.

Therefore, we can generate other sets of STF-compliant signals by shifting in time a compliant signal, or equivalently, using different dependency patterns $\Theta^{(\nu)} \triangleq \Theta_1 + \nu, \nu \in [-\pi, \pi]$. For example, Θ_2 in Table III can be represented by $\Theta^{(\pi/2)}$. A time shift, however, impacts the frame detection accuracy because the last few samples in the new STF signals can have higher amplitudes than the ones in the standardized STF. This can inflate the noise samples located before the true start of the frame when the autocorrelation window for frame detection includes a few of them. We exploit the LTF to remedy such errors.

We note that using different dependency patterns with the same φ is a form of frequency-domain differential PSK (FD-DPSK), which is robust to channel phasor and FO estimation errors (i.e., a non-coherent scheme). The number of different dependency patterns, denoted by Q, depends on the target performance of the demodulator (Section IV-B) and the frame detection accuracy (Section V-B). With the same φ, we can encode $\log_2 Q$ bits. See the example in Table III for $Q = 4$. In *P-modulation*, the assignment of bits to patterns and phase shifts follows the Gray Coding rule.

Altogether, the Tx can embed a total of $\log_2 M + \log_2 Q$ bits in the STF by simultaneously using the proposed time shift and phase rotation techniques. These techniques maintain the period, R_{PAP}, and R_{DR} of the standard STF, but change (1) the phase difference between successive frequency-domain symbols of an STS, and (2) the phase of the first frequency-domain symbol in the identical STSs. The following expression is a representation of the entire bit sequence that would be modulated using *P-modulation*, where b_i denotes the ith bit:

$$\Big[\underbrace{b_{\log MQ}, \ldots, b_{1+\log Q}}_{\varphi}, \underbrace{b_{\log Q}, \ldots, b_1}_{\nu}\Big]. \tag{5}$$

The bit sequence consists of all the bits that are modulated by rotation (parameter φ) and the dependency pattern (parameter ν). While in general the bits can appear in any order, in this representation we group the bits depending on the technique that is used to modulate them.

B. Sequence Demodulation

The process of message extraction involves detecting the most probable symbol sequence carried on the 12 subcarriers of the STF. In contrast to digital demodulation, in which the most probable symbols are independently and sequentially identified, in here a sequence of symbols should be detected simultaneously.

To demodulate and extract the embedded bit sequence, the Rx exploits the correlation among the symbols that belong to the same STS as well as the repetition of the STSs. It needs a small memory to first store the $10 \times 16 = 160$ received STF samples. After compensating for the FO and equalizing the effect of the estimated CSI on these samples, the Rx applies the 16-point discrete Fourier transform (DFT) on each of the STSs and obtains the symbols on each subcarrier. Then, it estimates ν. Let $\tilde{\mathcal{S}}_l = [\tilde{s}_{l,-6}, \ldots, \tilde{s}_{l,-1}, \tilde{s}_{l,1}, \ldots, \tilde{s}_{l,6}], l \in \{1, \ldots, 9\}$[4] be the sequence received on the lth STS and $\tilde{\nu}$ be the most probable estimate of ν based on $\tilde{\mathcal{S}}$. Because the elements in $\mathcal{S}$ are correlated, we use minimum mean-square error (MMSE) to estimate $\tilde{\nu}$ with respect to Θ_1. An example drawn from our experiments is shown in Fig. 4. Next, to find an estimate of φ, denoted by $\tilde{\varphi}$, the Rx subtracts the phases of the reference elements constructed according to $\Theta^{(\tilde{\nu})}$ from the corresponding elements in $\tilde{\mathcal{S}}_l$, $l = 1, \ldots, 9$. To account for noise when measuring the phase, the Rx calculates the sum of all $12l$ elements. The estimation of ν requires accurate frame detection while the estimation of φ needs accurate FO and CSI estimation. After FO correction and channel equalization, the bit sequence is fully demodulated based on $\tilde{\nu}$ and $\tilde{\varphi}$.

The additional computational complexity imposed on the Rx for sequence demodulation is $O(m \log m)$, where m is the number of received STF samples. Because $m = 160$ and is constant, we can conclude that the complexity is $O(1)$. The complexities of the individual tasks are as follows:

- Compensating for the estimated FO and channel equalization: $O(m)$
- Applying FFT/DFT on each of the 10 STSs: $O(m \log m)$
- Computing MSE for each possible value ν and finding the minimum MSE: $Q \times O(m)$
- Subtracting the phases and summing up the elements for estimating φ: $O(m)$

To study the limit on the order of the PSK modulation used for constellation rotation (i.e., the minimum phase by which a sequence is shifted) and the number of different time shifts Q (equivalently, the number of different Q-DPSK patterns), we set as a reference point the performance of the most reliable modulation scheme supported by the system (BPSK), which is often used for transmitting the PHY header. This will guarantee that if a frame is not discarded due to a decoding failure at PHY header when *P-modulation* is not employed, then such a frame is unlikely to be discarded when *P-modulation* is implemented.

The performance of demodulating a sequences of modulated symbols depends on the signal-to-noise ratio (SNR), FO and CSI estimation accuracy, and frame detection accuracy. In this section, we assume an AWGN channel as well as accurate FO and frame detection to obtain the M and Q that result in the same BER as BPSK. Let B_2 be the BER of 2-PSK (BPSK). Then, $B_2(\gamma) = Q\left(\sqrt{2\gamma}\right)$, where $Q(.)$ is the Q-function and γ denotes the SNR. The Rx first estimates $\Theta^{(\nu)}$ followed by φ.

1) Pattern Detection: The pattern $\Theta^{(\nu)}$, modulated using FD-DPSK, is the basis for estimating the other parameter φ. Let the BER of frequency-domain Q-DPSK modulation in *P-modulation* be $B_Q^D(\acute{\gamma})$, where $\acute{\gamma}$ is the effective SNR based on γ (calculated below). When the channel is flat-flat fading, the BER of DPSK is the same whether DPSK is time-domain or frequency domain [36]. However, FD-DPSK performs worse than the time-domain DPSK when the channel is frequency-selective/time-dispersive [36]. Assuming Gray Coding mapping, the expression of the exact BER over an AWGN channel is provided in [37, Eq. 8.86], but it does not have a closed-form for $M \geq 4$. We use the MATLAB function *berawgn*, which provides a very close approximation to the

[4] We cannot rely on all ten STSs because the last one may not be the same as other STSs due to pulse shaping at its boundary with the LTF.

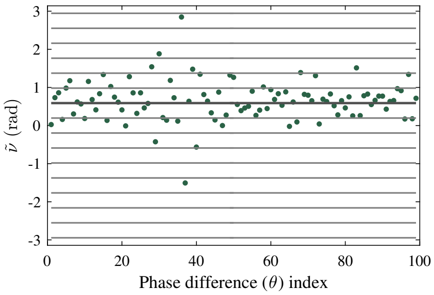
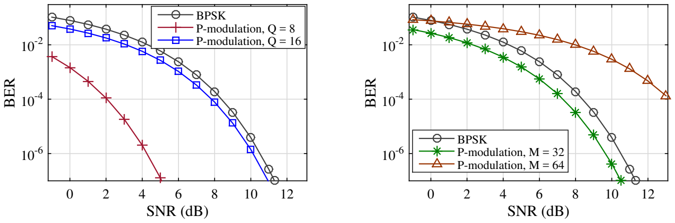


Fig. 4. 99 phase differences among $12 \times 9 = 108$ symbols extracted from 9 STSs. The Tx uses 16-DPSK and the Rx detects $\tilde{\nu} = 3\pi/16$ using MMSE estimator ($\gamma = 3$ dB).



(a) Q-DPSK (b) M-PSK

Fig. 5. BER of BPSK and *P-modulation* with different Q-DPSK and M-PSK schemes vs. SNR (γ).

exact BER [38], to sequentially search for the maximum Q that satisfies $B^D_Q(\acute{\gamma}) \leq B_2(\gamma)$. (DPSK modulation scheme loses 3 dB in gain compared to PSK due to the subtraction of two random variables.)

The amplitude of a symbol in the STF is $\sqrt{13/3}$ times larger than the one of a BPSK-modulated symbol in the frame payload. In addition, the duration of $l = 9$ STSs corresponds to the duration of $\frac{9}{4}$ OFDM symbols in the payload. However, because every two adjacent θ_i's are negatively correlated (an increase in one results in a comparable decrease in the other, see Fig. 4), using 11 DPSK symbols gives only $\frac{11}{2}$ SNR gain. Altogether, the SNR is improved by $\frac{13}{3} \times \frac{9}{4} \times \frac{11}{2} \approx 54$, i.e., $\acute{\gamma} \approx 54\gamma$. In Fig. 5(a), we compare $B^D_8(.)$ and $B^D_{16}(.)$ to $B_2(.)$. Even though $B^D_{16}(\acute{\gamma})$ is slightly less than $B_2(\gamma)$, the DFT error contributes to more BER when $Q = 16$, as will be shown in Section VII. Therefore, for an AWGN channel we set $Q = 8$. In the experiments, however, we revert to $Q = 16$ because of the robustness of DPSK to FO and CSI estimation errors.

2) Phase Detection: After performing $\Theta^{(\tilde{\nu})}$ estimation, FO correction, and channel equalization, the Rx estimates $\tilde{\varphi}$ using $12l$ symbols. Although one can arbitrarily increase the modulation order and reduce the distance between the PSK symbols without violating the STF requirements, we intend to maintain the BER below that of BPSK. Let the BER of an M-PSK modulation scheme in *P-modulation* be $B_M(\acute{\gamma})$. We want to find the maximum M such that $B_M(\acute{\gamma}) \leq B_2(\gamma)$.

Similar to the previous case, the amplitude of a PSK-modulated symbol in the STF is $\sqrt{13/3}$ times the one of a BPSK-modulated symbol. The summation of 12 i.i.d normally distributed random variables in frequency domain improves the SNR by 12 times. The SNR is further enhanced by using $l = 9$ repetitions of the same STS in time domain, which result in a duration (power) that is 2.25 times the duration (power) of an OFDM symbol in the frame. Therefore, the overall SNR improvement will be $\frac{13}{3} \times 12 \times 2.25 = 117$ and so $\acute{\gamma} = 117\gamma$.

The BER expression for PSK is provided in [37, Eq. 8.31]. It includes definite integrals but with integrands that involve exponentiation. Because an explicit expression for those integrals must be obtained for each value of M, a closed-form BER expression does not exist when $M > 4$ [37, p. 233]. Instead, a reasonably accurate approximation of $B_M(\gamma)$ that applies to both low and high SNRs and all M's is given by [37, Eq. 8.32]:

$$B_M(\gamma) \cong \frac{2}{\max(\log_2 M, 2)} \sum_{i=1}^{\max(\frac{M}{4},1)} Q\left(\sqrt{2\gamma \log_2 M} \sin\frac{(2i-1)\pi}{M}\right). \tag{6}$$

Numerically solving for the maximum M that satisfies

$$B_M(\acute{\gamma}) \leq Q\left(\sqrt{2\gamma}\right) \tag{7}$$

we obtain $M = 2^5 = 32 \ \forall\gamma \in \mathbb{R}$, and $M = 2^6 = 64 \ \forall\gamma \leq 0.2$ dB (see Fig. 5(b)).

C. Noncompliant but Possible STF Waveforms

In the scheme above, all the 802.11a STF requirements, including $R_{\text{PAP}} \leq 2.24$ dB and $R_{DR} \leq 7.01$ dB, are met. However, modern wireless devices are capable of processing signals with higher R_{PAP} and R_{DR} values. For example, COTS wireless routers usually support $R_{DR} > 100$ dB. This paves the way to expand the set Θ and include several new patterns by identifying STF waveforms whose R_{PAP} and R_{DR} values are close to the standard values but not identical to them, i.e., almost-compliant sequences. For example, by allowing R_{PAP} to increase to 2.95 dB, two new independent sets of patterns become available in addition to the set $\Theta^{(\nu)}$ defined in Section IV-A.

Moreover, by expanding the search space for s_i's of STF-compliant signals beyond QPSK, one may find other types of dependency patterns. Considering 8-PSK, for example, we identified at least one new pattern that is independent of the patterns that define the sequences with only QPSK symbols. For this pattern, $R_{\text{PAP}} \leq 2.27$ dB and $R_{DR} \leq 12$ dB, which are close to the standard values.

V. Effects of Channel/Device Impairments on P-modulation

The sequence demodulation process explained above is impacted not only by noise, but also by the complex channel coefficient h, residual FO estimation, and frame detection errors. In this section, we study the effect of each of these parameters and discuss the robustness of time shift and constellation rotation techniques against them. We further discuss why channel coding can actually harm *P-modulation*, rather than enhancing its potential for embedding more bits. Let δ_f and t_E represent the FO (normalized to Δ_f) and the timing error (in number of samples), respectively.

A. Tx-Rx Channel Coefficients

First, we consider the effect of the channel on the dependency pattern $\Theta^{(\nu)}$. Let h_k, $k = -6, \ldots, 6$ $(k \neq 0)$ be the time-invariant channel coefficient on the kth subcarrier during the STF. Assuming $\delta_f = 0$ and $t_E = 0$, symbol s_k on the lth STS will be received as $\tilde{s}_{l,k}$, i.e.,

$$\tilde{s}_{l,k} = h_k s_k + n_{l,k} \tag{8}$$

where $n_{l,k}$ is the noise component.

To detect the dependency pattern, the Rx calculates the phase difference between each pair of adjacent subcarriers. If the channel coherence bandwidth is larger than the frequency spacing between two adjacent subcarriers, the h_k's of every two adjacent subcarriers will have the same phase value and the channel does not impact the phase differences θ_i. For each set of subcarriers with the same h_k's, the Rx computes θ_i's separately. On the other hand, if the coherence bandwidth is smaller than $4\Delta_f$ and the phases of h_k's become uncorrelated, the variations will be averaged out when MMSE over 108 symbols is used. The same argument can be made for multi-path channels. Furthermore, because the proposed demodulator treats STSs independently and the coherence time is larger than λ_S, θ_i's are not affected if the channel is fast fading/time-selective

Similarly, when $\delta_f \neq 0$, the subcarriers that belong to the same STS will experience the same amount of phase offset and so residual FOs do not impact the decoding performance. However, FD-DPSK is susceptible to timing errors. We remedy that by taking advantage of the LTF (see below).

Although the channel has little impact on $\tilde{v}$, it may significantly change the phase of the PSK symbols in $\mathcal{S}$ by a constant, i.e., the channel phasor. So the Rx needs to estimate the channel and equalize it before estimating φ.

B. Frame Detection Accuracy

A time offset manifests itself as a linear phase shift in the frequency domain [35], [39]. This phase shift varies linearly with the subcarrier index and may significantly change the phase differences between successive symbols in $\mathcal{S}$. A timing error t_E creates phase shift ν_k in the kth subcarrier:

$$\nu_k = \frac{2\pi k t_E}{N}, k = -N/2, \ldots, N/2. \tag{9}$$

Fig. 6 depicts an example (drawn from our experiments) of the linear change of (wrapped) estimated channel phases with the subcarrier frequency when frame detection is not accurate. A linear phase shift in the frequency domain will itself result in a constant phase change $2\pi t_E/N$ in the pattern Θ_i. For example, if $t_E = 4$, $N = 16$, and Θ_1 in Table III are used for each STS, the Rx will detect Θ_2 because $\Theta_2 = \Theta_1 + \pi/2$. Hence, correct detection of the pattern depends on accurate detection of the STF.

In *P-modulation*, we take advantage of already available time-domain LTF-based CSI estimation to fine-tune the frame detection. In this technique, the Rx usually considers a channel estimation period of length, say L taps, to estimate the maximum of L multi-path channel components. Once the STF has been detected, the Rx constructs an L-row *Toeplitz* matrix whose first row corresponds to the known LTF and the remaining $L-1$ rows are filled with shifted versions of the LTF. Using an MMSE estimator and the Toeplitz matrix, the Rx then estimates the channel coefficient of each path. Assuming that the signal coming along the first path is the strongest one (e.g., LOS), the Rx estimates t_E by finding the highest coefficient amplitude among the channel coefficients.

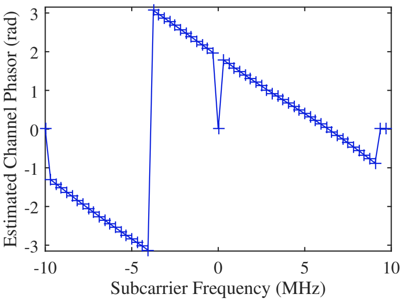


Fig. 6. Linear channel phase changes across OFDM subcarriers ($t_E = -4$, $N = 64$, and Tx power = -25 dBm).

Due to the time shift, however, the magnitude of the last STS sample may become smaller than the one in the default STS. This can result in more frequently detecting the frame before its true start time [12]. To cover such cases, the Rx may extend L by an integer L', enlarging the size of the Toeplitz matrix. Hence, irrespective of *P-modulation*, this technique would involve L' additional CSI estimation rounds beyond the rounds that are performed by classical LTF-based CSI estimation. The MMSE estimation involves matrix multiplications and inversions over the Toeplitz matrix. Now if the complexity of the MMSE in the default mode is $O(L^3)$, the complexity of CSI estimation in *P-modulation* would be $O((L+L')^3)$, which is essentially the same as $O(L^3)$. Therefore, the additional burden of our fine-synchronization method is negligible.

The above method is more reliable than the fine-synchronization method in the SourceSync protocol [39], which is performed before CSI estimation. In SourceSync, the Rx uses the phase of the estimated channel coefficients $h_{l,k}$ to find $2\pi t_E/N$, i.e., the slope of the phase values ν_k with respect to the subcarrier index k. However, because *P-modulation* shifts the STF in time as part of its design, applying the synchronization method of SourceSync deteriorates both the frame detection accuracy and DPSK demodulation. In spite of that, SourceSync can be applied to further fine-tune the frame estimation after LTF-based frame detection. In this case, any (residual) timing error less than $\lambda_S/(NQ/4)$ can be estimated if the frame detection is applied on an oversampled version of the signal. Note that the minimum time shift t_s caused by the Q patterns is equal to $\frac{2\lambda_S}{NQ/4}$. When LTF-based CSI estimation uses a downsampled version of the signal, this method takes advantage of the estimated pattern $\Theta^{(\tilde{\nu})}$ to estimate the amount of extra phase shift in the STSs with respect to θ_i's of $\Theta^{(\tilde{\nu})}$. Assuming that the oversampling parameter is known, this method can unambiguously estimate the timing error using (9) as long as the extra phase shift is less than $\frac{2\pi t_E}{2Q}$. The Rx will compensate for this error before estimating φ.

C. Frequency Offset Estimation Error

FO at the Rx moves all the subcarriers in the frequency domain by δ_f. Moving away from the expected frequency locations of the subcarriers changes the phase and amplitude of a

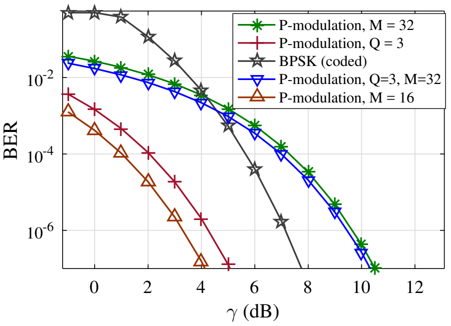


Fig. 7. BER of channel-coded BPSK and uncoded *P-modulation* under different Q-DPSK or M-PSK schemes vs. SNR (γ).

symbol and also creates inter-carrier interference, as explained below (assume for simplicity $h_k = 1$ and $t_E = 0$) [40]:

$$\tilde{s}_{l,k} = T_0 s_k + \sum_{i=-6, i\notin\{0,k\}}^{6} T_{i-k} s_i + n_{l,k}, k = -6, \cdots, 6 \qquad (10)$$

where

$$T_k \triangleq \frac{\sin \pi(k+\delta_f)}{N \sin \frac{\pi}{N}(k+\delta_f)} \exp\left[j\pi\left(1-\frac{1}{N}\right)\left(k+\delta_f\right)\right], k = -6, \ldots, 6. \qquad (11)$$

We can observe from the equations above that: (1) the phases of s_k's change by the same value $\measuredangle(T_0)$, and (2) a weighted sum of other symbols contributes to $\tilde{s}_{l,k}$. With respect to the first observation, FO does not impact the phase difference between the symbols in $\mathcal{S}$; and hence, the underlying pattern. The linear phase change, however, affects the estimation of φ. With respect to the second observation, we note that $\delta_f \ll 1$ after both STF- and LTF-based FO corrections and hence T_k will be close to zero.

D. Channel Coding

In existing systems, channel coding is used to reduce the BER below a desired threshold at the expense of lowering the data rate and increasing the transmission duration. There is, however, an essential difference between coding in these systems and coding in *P-modulation*. In existing systems, the same set of uncoded bits can be mapped to different codewords of different sizes, depending on the code rate. The coded bits are then modulated using a channel-dependent modulation scheme, whose modulation order is independent of the number of bits in the codeword. Block and convolutional codes are two major classes of such codes. In contrast, in *P-modulation* all the bits are modulated using only two symbols, one M-PSK and one Q-DPSK symbol. The number of (uncoded) bits specify the "order" of the modulated symbols, i.e., parameters M and Q, irrespective of the channel condition. The desired BER performance determines the number of bits. As discussed above, in *P-modulation* the desired BER performance is set to that of BPSK, the most reliable modulation scheme, to make sure that *P-modulation* has an error probability less than that of the PHY header. In this case, the probability of packet loss due to errors in the preamble will never be higher than the packet loss probability in existing systems.

Considering the fact that the PHY header in existing systems 802.11 system is often coded before modulation, we need to compare the BER performance of uncoded *P-modulation* with the performance of BPSK with coding. For coded BPSK, we consider the most reliable coding used in 802.11a [1], which uses a convolutional encoder based on generator polynomials $g_0 = 133_8$ and $g_1 = 171_8$, and code rate $R = 1/2$. We see in Fig. 7 that *P-modulation* with only Q-DPSK when $Q = 3$ (3 bits) is even more reliable than coded BPSK. That is also the case for M-PSK with $M = 16$ (4 bits). Likewise, 32-PSK is more reliable than coded BPSK when $\gamma \leq 4$. So modulating the total of 8 bits using combined 3-DPSK and 32-PSK can perform better than coded BPSK when $\gamma \leq 5$. It is also clear from the figure that P-modulating 7 bits using 3-DPSK and 16-PSK always achieves better BER than coded BPSK.

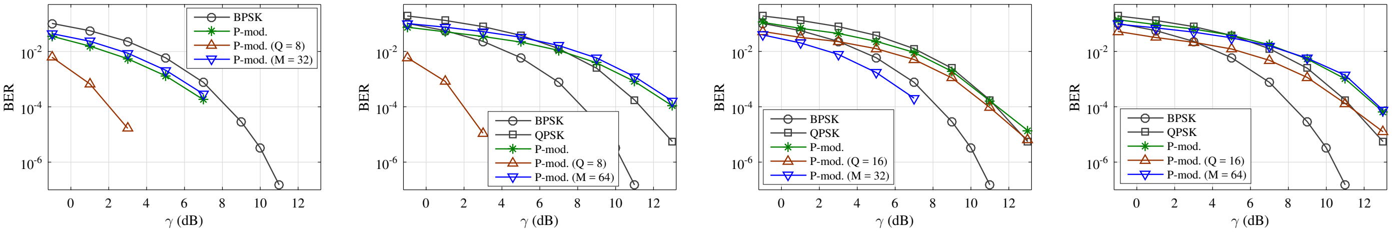
Although one may consider applying block coding[5] to reduce the BER and increase the number of bits that *P-modulation* can embed in the STF, we argue that coding is not necessarily beneficial in *P-modulation*. Block coding increases the number of bits, leading to larger M and Q. Very often, the increase in the error probability due to the increase in the modulation order in *P-modulation* outweighs the gain of coding. We have verified this using the coding functions of MATLAB (the results are not shown here). Note that when the *block-coded* bits are all modulated using a single symbol, the number of distinct symbols would be the same as the number of lower-order symbols used to modulate the corresponding *uncoded* bits. So in PSK and DPSK, the BER of coded bits will never be lower than the BER of uncoded bits.

VI. MIMO and 802.11n/ac Systems

The aforementioned STF structure with 12 subcarriers was defined for 802.11 OFDM systems that operate over a 20 MHz channel. In 802.11n and 802.11ac systems, the preambles are essentially the same as in 802.11a, but with certain modifications to customize them for channel bandwidths up to 160 MHz. Therefore, we can easily use *P-modulation* for embedding a bit sequence in the preamble of 802.11n/ac systems, as follows.

With the same subcarrier spacing Δ_f defined in 802.11a, the STF in 802.11n and 802.11ac can have up to 48 [2] and 96 [3] subcarriers, respectively. The same 802.11a preamble, which spans 20 MHz bandwidth, is duplicated and rotated by 90° for the other segments of the bandwidth. Hence, the length of the sequence $\mathcal{S}$ is increased proportionally, and so the SNR. Accordingly, the Tx can increase M and Q, and embed more bits in the STF waveform. For example, with 160 MHz channel, *P-modulation* can increase the embedding capacity by at least one bit using the time shift technique and another bit using the phase rotation technique (a total of two bits). Moreover, the preamble of the mandatory HT-mixed format of 802.11n for MIMO systems contains a second STF with half the length of the first one. Similar to the first STF, we can embed a bit sequence (with one bit less than the first sequence) in the second STF and almost double the length of the embedded sequence (achieving a total of up to 19 bits).

[5] Convolutional coding is likely not practical here because the number of bits is limited and so a decoder cannot take advantage of a sufficiently large sequence of received symbols for decoding. A more rigorous study of the applicability of convolutional codes is left for future work.



(a) $Q = 8$ and $M = 32$ (b) $Q = 8$ and $M = 64$ (c) $Q = 16$ and $M = 32$ (d) $Q = 16$ and $M = 64$

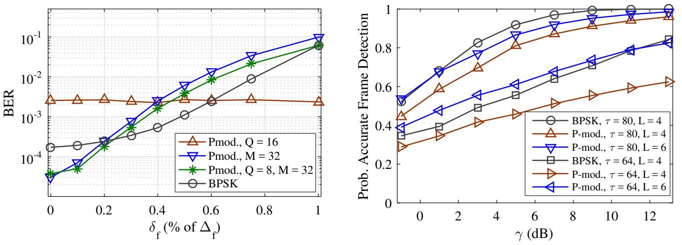
Fig. 8. BER vs. γ (simulations). For P-mod. (FD-DPSK) and P-mod. (PSK), only the value of Q and M applies, respectively.

VII. Performance Evaluation

In this section, we study the performance of *P-modulation* through simulations and indoor USRP experiments. The whole scheme was implemented in LabVIEW. The Tx embeds 8-10 randomly generated bits ($Q = 8$ or 16, and $M = 16$, 32, or 64) in the STF. For simplicity, the preamble is transmitted without appending any payload. As a convention, we refer to *P-modulation* that only uses time-shifted STF waveforms (via FD-DPSK) as "P-mod. ($Q = \dots$)". Similarly, "P-mod. ($M = \dots$)" refers to *P-modulation* with phase-rotated waveforms at a given M value. Finally, "P-mod." refers to the complete scheme with given Q and M values. The metrics of interest are frame detection and FO estimation accuracy as well as BER. We vary the received SNR (γ) and the residual FO estimation error (δ_f). We also implemented BPSK and QPSK modulation schemes as benchmarks for comparison with *P-modulation*. In this case, the Tx appends a 120-byte payload (equivalent to 10 or 20 OFDM symbols) with 12.5% cyclic prefix to the preamble and transmits it. The bit sequences are all uncoded.

A. Simulations

First, we study *P-modulation* assuming perfect time and frequency synchronization under an AWGN channel model. In Fig. 8, we compare the BER performance of *P-modulation* with different combinations of M and Q against the performance of BPSK and QPSK. Each point is obtained by averaging the BER over 40000 iterations. When 8 bits are embedded in the STF (Fig. 8(a)), our proposed scheme outperforms BPSK and confirms the analytical analysis in Section IV-B. However, although increasing M to 64 keeps the overall BER below that of QPSK at low SNR (i.e., $\gamma < 7$ dB), it becomes worse than both BPSK and QPSK at high SNR (Fig. 8(b)). The reason is the intrinsic difference in the BER trends (i.e., slopes) of BPSK/QPSK and higher-order PSK. The BER of QPSK decreases faster than that of 64-PSK. If we instead increase Q to 16 and embed 9 bits, Fig. 8(c) shows that the overall performance will be comparable to QPSK, irrespective of the SNR value. Finally, Fig. 8(d) shows that *P-modulation* can embed 10 bits and achieve the performance of QPSK when $\gamma < 5$ dB.



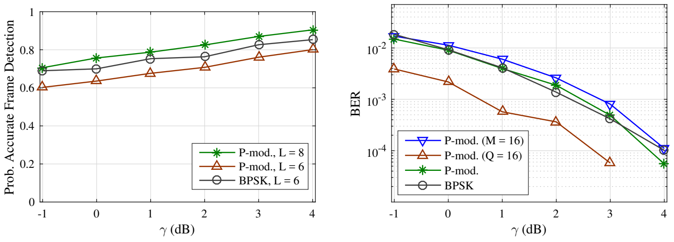
(a) BER vs. residual δ_f (b) Frame detection accuracy vs. γ ($M = 32$ and $Q = 16$)

Fig. 9. Robustness of *P-modulation* to channel impairments compared to BPSK (simulations).

Next, in Fig. 9, we study the robustness of *P-modulation* to residual δ_f in decoding and to noise in frame detection. Fig. 9(a) shows that the proposed DPSK-based embedding scheme is independent of the residual δ_f. A given δ_f rotates the phases of the symbols in $\mathcal{S}$ by the same amount. However, such a phase change is cancelled out when the phase differences are calculated. So when $\delta_f > 0.6$, 16-DPSK outperforms BPSK (8-DPSK always performs better than BPSK, but it is not shown here). In contrast, *P-modulation* with M-PSK is very sensitive to FO estimation errors. With $M = 32$, it can maintain its superiority over BPSK only when $\delta_f < 0.2$. In spite of that, we observe that when the Tx embeds 8 bits in the STF using $Q = 8$ and $M = 32$, the overall BER is lower than BPSK when $\delta_f \leq 0.24$. This improvement is due to the contribution of the 3 bits that are embedded using 8-DPSK in the overall BER. We also measured the FO estimation accuracy. The results (not shown here due to space limitations) confirm that the FO estimation performance is not impacted by *P-modulation* because the set of sample amplitudes in *P-modulation* is the same as in the standardized STF, although they are ordered differently.

Finally, in Fig. 9(b) we consider the implications of *P-modulation* on frame detection. The results in this figure include the cases when the Rx uses the whole STF for frame detection ($\tau = 80$) as well as the cases when only the first eight STSs are used ($\tau = 64$). As discussed in Section IV, the time shifts in the STF due to using different patterns degrade the performance of STF-based frame detection. This is mainly attributed to the fact that the last two samples of the new STSs may have a high amplitude. Hence, when the autocorrelation window is placed one or two time instances before the true frame start and the noise terms are multiplied by those samples of one of the STSs, $\mathcal{A}_\tau$ can be very close to the $\mathcal{A}_\tau$ computed at the true start. So we include two more of the samples that are before the true frame start by setting $L = 6$ instead of $L = 4$. Fig. 9(b) shows that the frame detection accuracy becomes comparable to the one when the standardized STF is used with $L = 4$. Therefore, if the Rx extends its channel estimation length by two, it can remedy the implications of *P-modulation* on frame detection.

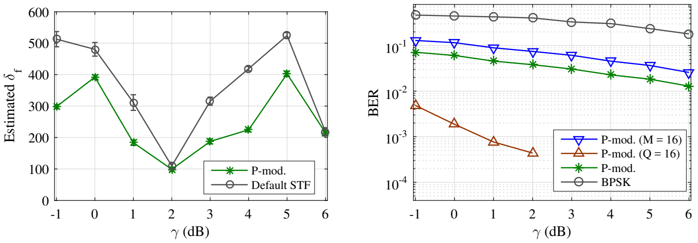
Altogether, the results in Fig. 9 suggest a trade-off between the number of bits embedded using Q-DPSK ($\log_2 Q$) and the



(a) Frame detection accuracy

(b) BER

Fig. 10. Performance of *P-modulation* vs. γ when $Q = 16$, $M = 16$, and $\delta_f \approx 0$ (USRP experiments).



(a) Average of estimated δ_f

(b) BER

Fig. 11. Performance of *P-modulation* vs. γ when $Q = 16$ and $M = 16$, and under imperfect FO estimation (USRP experiments).

number of bits embedded using M-PSK ($\log_2 M$), depending on the accuracy of the FO estimation relative to the frame detection accuracy. That is, when the frame detection accuracy is good, the Tx can increase Q; when the FO estimation is accurate, the Tx can increase M.

B. USRP Experiments

We also implemented *P-modulation* on a USRP testbed that consists of two NI-USRP 2922 devices, placed 2 meters from each other. The Tx and the Rx each uses a 3-dBi antenna. The minimum transmission power of the Tx is −8 dBm. However, because this power is still very high with respect to the noise floor (< −80 dBm), we first add artificial Gaussian noise to the generated signal and then apply an artificial channel gain of −14 dB before the transmission takes place. In here, we vary the SNR at the Tx. Furthermore, to distinguish the impact of the channel from that of the FO estimation errors, we use an Ettus OctoClock-G clock distributer to share an external clock for the USRPs and eliminate the FO. Due to the robustness of FD-DPSK to CSI and FO estimation errors, the Tx embeds 8 bits using $Q = 16$ and $M = 16$. The Rx uses the channel estimation of length $L = 6$ for BPSK and $L = 8$ for *P-modulation*. Frame detection lag parameter τ is set to its maximum and the signal is transmitted in the 2.5 GHz band to avoid external WiFi interference.

Fig. 10 shows the performance of *P-modulation* in our USRP experiments when the radios share a clock. Extending the channel estimation length by two samples ($\lambda_S/8$ seconds) pays off and the frame detection accuracy of *P-modulation* statistically matches the one of the default STF, as shown in Fig. 10(a). With respect to the BER performance, we observe in Fig. 10(b) that CSI estimation errors contribute to the high BER in the bits embedded by phase rotation even though M is set to 16. (Each point is obtained by averaging the BER over 5000 transmissions.) However, the bits modulated using 16-DPSK are very robust to such errors, even more than BPSK. The low BER of these bits makes the overall performance of the 8-bit *P-modulation* and BPSK comparable.

Finally, we disconnect the OctoClock and evaluate the BER and FO estimation performance of *P-modulation* in Fig.11. With respect to the estimation of the (time-varying) FO, Fig. 11(a) shows that the overall accuracy of STF- and LTF-based FO estimation in *P-modulation* does not have a meaningful difference compared to the default STF. The difference between the two curves is attributed to the fact that the FOs of the USRPs are time-varying and so are slightly changed from the time we ran the experiment for *P-modulation* to the time we ran it for the default STF. Furthermore, we observe in Fig. 11(b) that the residual FO significantly deteriorates the performance of 16-PSK, but 16-DPSK exhibits a high robustness to such errors. Nevertheless, *P-modulation* in this case has a much lower BER than BPSK.

VIII. Conclusion and Future Directions

In this paper, we took a crucial step towards enabling several new PHY-layer functions in OFDM-based 802.11 systems by embedding user-generated bit sequences in the STF of the preamble of these systems. The envisioned applications include enhancing the security at the PHY layer or providing a PHY-layer signalling for improved performance. To do so, we proposed *P-modulation* in which a set of new STF waveforms are constructed that comply with the requirements of the preamble of OFDM-based IEEE WLAN systems. These compliant waveforms are obtained using two techniques: shift in the time domain and phase rotation in the frequency domain. We then modulated a bit sequence using these two techniques. For the same (or better) BER performance of BPSK modulation, our analysis indicates that the transmitter can embed up to 8 bits using *P-modulation* when operating on 20 MHz bandwidth. If higher-bandwidth channels are used (e.g., in 802.11n/ac systems), more bits can be embedded. Our simulation and USRP experiment results confirm the practicality of *P-modulation* in real scenarios.

One of the design goals of *P-modulation* is to maintain backward-compatibility with existing 802.11 devices. However, our design techniques are not specific to existing 802.11 systems and can be employed in future systems as long as the preamble has a repetitive structure. In fact, we showed that relying on a repetitive structure to design the preamble instead of relying on the exact preamble waveform creates a capacity to embed function-specific bits, while maintaining the required preamble functions. In addition, by exploiting the recent advances in RF design, one can relax the tight constraints of the preamble and significantly improve its embedding capacity.

References

[1] "IEEE Std 802.11a-1999," *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, 1999.

[2] "IEEE Std 802.11n-2009," *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, 2009.

[3] "IEEE Std 802.11ac-2013," *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz*, 2013.

[4] H. Rahbari and M. Krunz, "Full frame encryption and modulation obfuscation using channel-independent preamble identifier," *IEEE Trans. Inf. Forens. Security*, vol. 11, no. 12, pp. 2732–2747, Dec. 2016.

[5] V. Kumar, J.-M. Park, and K. Bian, "Blind transmitter authentication for spectrum security and enforcement," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, Scottsdale, Arizona, USA, 2014, pp. 787–798.

[6] Z. Feng *et al.*, "Coping with packet replay attacks in wireless networks," in *Proc. IEEE SECON Conf.*, Jun. 2011, pp. 368–376.

[7] T. C. Clancy, "Efficient OFDM denial: Pilot jamming and pilot nulling," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2011, pp. 1–5.

[8] C. Shahriar, R. McGwier, and T. Clancy, "Performance impact of pilot tone randomization to mitigate OFDM jamming attacks," in *Proc. IEEE Consumer Commun. Netw. Conf. (CCNC)*, Jan. 2013, pp. 813–816.

[9] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Interleaving jamming in Wi-Fi networks," in *Proc. 9th ACM WiSec Conf.*, 2016, pp. 31–42.

[10] T. D. Vo-Huu and G. Noubir, "Mitigating rate attacks through crypto-coded modulation," in *Proc. 16th ACM MobiHoc*, 2015, pp. 237–246.

[11] H. Rahbari, M. Krunz, and L. Lazos, "Swift jamming attack on frequency offset estimation: The Achilles' heel of OFDM systems," *IEEE Trans. Mobile Computing*, vol. 15, no. 5, pp. 1264–1278, 2016.

[12] H. Rahbari and M. Krunz, "Rolling preambles: Mitigating stealthy FO estimation attacks in OFDM-based 802.11 systems," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, Oct. 2016.

[13] T. Stöber, M. Frank, J. Schmitt, and I. Martinovic, "Who do you sync you are? smartphone fingerprinting via application behaviour," in *Proc. 6th ACM WiSec Conf.*, Budapest, Hungary, 2013, pp. 7–12.

[14] Q. Wang, A. Yahyavi, B. Kemme, and W. He, "I know what you did on your smartphone: Inferring app usage over encrypted data traffic," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, Sep. 2015, pp. 433–441.

[15] H. Rahbari and M. Krunz, "Secrecy beyond encryption: obfuscating transmission signatures in wireless communications," *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 54–60, 2015.

[16] G. Noubir, R. Rajaraman, B. Sheng, and B. Thapa, "On the robustness of IEEE 802.11 rate adaptation algorithms against smart jamming," in *Proc. 4th ACM WiSec Conf.*, Hamburg, Germany, Jun. 2011, pp. 97–108.

[17] R. Savage, "Snooping garbage bins in city of London ordered to be disabled," Aug. 2013. [Online]. Available: http://goo.gl/omktFa

[18] C. J. Bernardos, J. C. Zúñiga, and P. O'Hanlon, "Wi-Fi Internet connectivity and privacy: hiding your tracks on the wireless Internet," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Tokyo, Japan, Oct. 2015.

[19] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis," *Mobile Netw. Applicat.*, vol. 10, no. 3, pp. 315–325, 2005.

[20] F. Armknecht, J. Girao, A. Matos, and R. L. Aguiar, "Who said that? privacy at link layer," in *Proc. IEEE INFOCOM*, Anchorage, Alaska, USA, May 2007, pp. 2521–2525.

[21] B. Greenstein *et al.*, "Improving wireless privacy with an identifier-free link layer protocol," in *Proc. 6th Int. Conf. Mobile Syst., Appl., and Services*, Breckenridge, CO, USA, 2008, pp. 40–53.

[22] Y. Fan, B. Lin, Y. Jiang, and X. Shen, "An efficient privacy-preserving scheme for wireless link layer security," in *Proc. IEEE GLOBECOM Conf.*, New Orleans, LO, USA, Nov. 2008, pp. 1–5.

[23] M. Vanhoef *et al.*, "Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms," in *Proc. 11th ACM Asia Conf. Comput. Commun. Security (ASIA CCS)*, Xi'an, China, Jun. 2016, pp. 413–424.

[24] Z. Jiang *et al.*, "Rejecting the attack: Source authentication for Wi-Fi management frames using CSI information," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 2544–2552.

[25] H. Liu *et al.*, "Practical user authentication leveraging channel state information (CSI)," in *Proc. 9th ACM Symp. Inf. Comput. Commun. Security (ASIA CCS)*, Kyoto, Japan, 2014, pp. 389–400.

[26] I. E. Bagci, U. Roedig, M. Schulz, and M. Hollick, "Short paper: Gathering tamper evidence in Wi-Fi networks based on channel state information," in *Proc. ACM WiSec Conf.*, Oxford, U. K., 2014, pp. 183–188.

[27] X. He *et al.*, "Toward proper guard zones for link signature," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2104–2117, Mar. 2016.

[28] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. 14th ACM MobiCom*, San Francisco, California, USA, 2008, pp. 116–127.

[29] S. U. Rehman, K. W. Sowerby, and C. Coghill, "Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers," *J. Comput. Syst. Sci.*, vol. 80, no. 3, pp. 591 – 601, 2014.

[30] "Specification framework for TGah," May 2013, doc.: IEEE P802.11-11/1137r15.

[31] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*. SAMS Publishing Indianapolis, 2002.

[32] R. Boehnke and T. Doelle, "Alternative proposal for BRAN SYNCH preamble," Mar. 1999, doc.: IEEE 802.11-99/048. [Online]. Available: http://goo.gl/NIy6BM

[33] J. Classen, M. Schulz, and M. Hollick, "Practical covert channels for WiFi systems," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2015, pp. 209–217.

[34] "P802.11a draft D5.5 comments," Jun. 1999, doc.: IEEE P802.11-99/1460. [Online]. Available: http://goo.gl/Rw3Zxi

[35] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems*. Prentice-Hall, 1996.

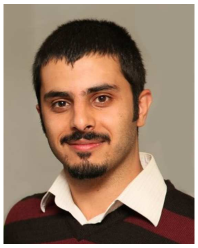
[36] K. Zhong, T. T. Tjhung, and F. Adachi, "A general SER formula for an OFDM system with MDPSK in frequency domain over Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 584–594, 2004.

[37] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*, 2nd ed. John Wiley & Sons, 2005.

[38] MATLAB, "Analytical expressions used in berawgn," 2015. [Online]. Available: https://goo.gl/A0L5X5

[39] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: a distributed wireless architecture for exploiting sender diversity," in *Proc. ACM SIGCOMM*, New Delhi, India, Sep. 2010, pp. 171–182.

[40] J. Armstrong, "Analysis of new and existing methods of reducing intercarrier interference due to carrier frequency offset in OFDM," *IEEE Trans. Commun.*, vol. 47, no. 3, pp. 365–369, Mar. 1999.



Hanif Rahbari received his B.Sc. degree in information technology from Sharif University of Technology, Iran, his M.Sc. degree with honors in computer networks from AmirKabir University of Technology, Iran, and his P.hD. degree in electrical and computer engineering from the University of Arizona in 2016. He is currently a postdoctoral associate at the Bradley Department of Electrical and Computer Engineering, Virginia Tech. His research interests include wireless networks and IoT, security and privacy issues in wireless communications, hardware experimentation, and dynamic spectrum access.

Marwan Krunz is the Kenneth VonBehren Endowed Professor in the Department of ECE, the University of Arizona, and the site co-director of the Broadband Wireless Access and Applications Center. He received his Ph.D. degree in electrical engineering from Michigan State University in 1995 and joined the University of Arizona, after a brief postdoctoral stint at the University of Maryland. His research interests lie in the areas of wireless communications and networking, with emphasis on resource management, adaptive protocols, and security issues. He has published more than 250 journal articles and peer-reviewed conference papers. M. Krunz is an IEEE Fellow, an Arizona Engineering Faculty Fellow, and an IEEE Communications Society Distinguished Lecturer (2013 and 2014). He was the recipient of the 2012 IEEE TCCC Outstanding Service Award and the NSF CAREER award. He has served and continues to serve on the editorial boards of several IEEE/ACM journals and currently serves as the Editor-in-Chief for the IEEE Transactions on Mobile Computing (TMC). He was the general vice-chair for WiOpt'16; general co-chair for WiSec'12; and TPC chair for WCNC'16 (Networking Track), INFOCOM'04, SECON'05, WoWMoM'06, and Hot Interconnects'9.