# Sampleless Wi-Fi: Bringing Low Power to Wi-Fi Communications

Wei Wang, *Member, IEEE,* Yingjie Chen, Lu Wang, *Member, IEEE,* and Qian Zhang, *Fellow, IEEE*

***Abstract*—The high sampling rate in Wi-Fi is set to support bandwidth-hungry applications. It becomes energy inefficient in the post-PC era in which the emerging low-end smart devices increase the disparity in workloads. Recent advances scale down the receiver's sampling rates by leveraging the redundancy in the physical layer (PHY), which, however, requires packet modifications or very high signal-to-noise ratio (SNR). To overcome these limitations, we propose *Sampleless Wi-Fi*, a standard compatible solution that allows energy-constrained devices to scale down their sampling rates regardless of channel conditions. Inspired by rateless codes, Sampleless Wi-Fi recovers under-sampled packets by accumulating redundancy in packet retransmissions. To harvest the diversity gain as rateless codes without modifying legacy packets, Sampleless Wi-Fi creates new constellation diversity by exploiting the time shift effect at receivers. Our evaluation using GNURadio/USRP platform and real Wi-Fi traces have demonstrated that Sampleless Wi-Fi significantly outperforms the state-of-the-art downclocking technique in both decoding performance and energy efficiency.**

***Index Terms*—Energy efficiency, Wi-Fi, Sub-Nyquist Sampling**

## I. Introduction

The dramatic increase in Wi-Fi speeds in recent years has promised to deliver sufficient bandwidth to support high data rate services such as high-definition video streaming and bulky file transfer. As we are entering the post-PC era in which an ever-larger variety of mobile devices such as smartphones and wearables are increasing the disparity between the types of traffic demands. While high speed Wi-Fi fits high-end devices well for their heavy workloads, it brings little benefit to low-end devices with light workloads but rather becomes an overwhelming energy burden. The need for low-power Wi-Fi increases as growing numbers of wearables have enabled Wi-Fi to directly connect to the Internet. For example, Google's latest Android Wear release [1] allows android smartwatches to connect to Wi-Fi without smartphones; Apple Watch is already equipped with a built-in Wi-Fi chipset and may enable direct Internet connectivity in future WatchOS releases. Unfortunately, the current crop of Wi-Fi designs are ill-suited for the coming wave of power-constrained smartphones and wearables, as they are primarily designed for high-end devices.

Growing attempts and extensive efforts have been devoted to providing energy efficient Wi-Fi connectivity to embrace the coming wave of energy-constrained smartphones and wearables. It has been shown that the primary cause of Wi-Fi's high power consumption comes from the high sampling rate of its radios [2], [3]. Recent advances have demonstrated the feasibility of downclocking the sampling rate of Wi-Fi radios while detecting [3] and receiving packets [4], [5] under high channel quality conditions. However, hardly any of them have made it to receive packets from today's Wi-Fi access points (APs). They either require modifications to legacy preambles [3], or excess redundancy at the physical layer (PHY) (direct sequence spread spectrum (DSSS) [4] or radically conservative data rates under high channel quality conditions [5]). Unfortunately, the current crop of Wi-Fi techniques have almost used up the redundancy at PHY by employing orthogonal frequency-division multiplexing (OFDM) and rate adaptation schemes.

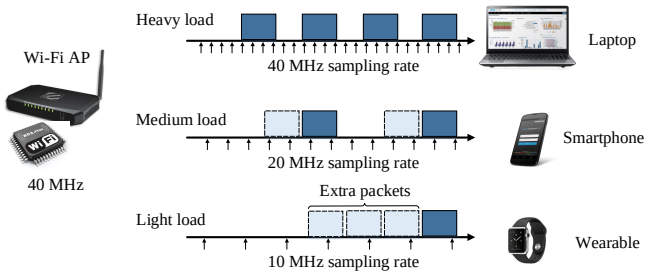



Fig. 1. Illustration of Sampleless Wi-Fi.

Instead of relying on PHY redundancy which is barely left in today's Wi-Fi networks, we argue that a more practical and controllable way is to exploit the media access control layer (MAC) that can be directly controlled on existing Wi-Fi access points (APs), while making minimum PHY changes only at downclocked receivers. As such, future mobile devices with downclocking capabilities can be seamlessly integrated into current Wi-Fi networks without requiring dedicated APs.

In this work, we propose *Sampleless*[1] *Wi-Fi*, which aims to provide reliable communications between legacy Wi-Fi APs and mobile devices at various sampling rates. Sampleless Wi-Fi is inspired by the wisdom of *rateless codes* [6] in that an undecodable packet in a single transmission can be recovered by combining multiple transmissions. In Sampleless Wi-Fi, in a same data transmission, effective packet reception with different energy consumption can be achieved for receivers at different sampling rates. The concept of Sampleless Wi-Fi

W. Wang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology and the Fok Ying Tung Research Institute, Hong Kong University of Science and Technology. E-mail: gswwang@cse.ust.hk.

Y. Chen, L. Wang, and Q. Zhang are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. E-mail: {ychenax, wanglu, qianzh}@cse.ust.hk.

[1]Sampleless implies that packets can be sampled at various rates regardless of data rates and channel conditions.

is illustrated in Fig. 1, where versatile devices with various workloads access the Internet via a Wi-Fi AP. A laptop communicates with the AP at the Nyquist sampling rate (40 MHz) to meet heavy traffic demands; while for less bandwidth-hungry mobile devices that use sub-Nyquist sampling rates (20 MHz or 10 MHz) for energy saving, the AP incrementally adds redundancy by sending extra packets until successful reception. As a result, the AP sends legacy packets without PHY modifications, and energy-constrained devices can down-clock their sampling rates without requiring specific channel conditions or PHY redundancy.

A key challenge in Sampleless Wi-Fi is how to add redundancy through multiple transmissions without modifying AP's PHY. Rateless codes generally require senders to transmit correlated symbols, which incurs baseband modifications that are not supported by today's APs. Departure from rateless codes, in Sampleless Wi-Fi, APs send repeated legacy packets through retransmissions, and receivers yield correlated symbols by exploiting the *time shift* effect. Specifically, Sampleless Wi-Fi receivers leverage the fact that one under-sampled symbol contains multiple subcarrier symbols due to *frequency aliasing*, and a time shift results in different phase rotations at different subcarriers. Thus, the receivers inject various time shifts for repeated packets to obtain different constellation symbols, and then combine these symbols for decoding.

Another challenge stems from reliable and efficient packet reception under low signal-to-noise ratio (SNR) conditions. Previous work [5] demonstrates the feasibility of receiving under-sampled OFDM packets under high SNR conditions (>25 dB), while reliable frame synchronization at low SNR remains challenging. Sampleless Wi-Fi jointly utilizes under-sampled preambles across repeated packets to achieve desirable correlation results. In addition, Sampleless Wi-Fi carefully designs the decoding algorithm to minimize overhead. It enables multiple retransmissions using Transmit Opportunity (TXOP) [7]. As a result, extra packets generated in Sampleless Wi-Fi do not increase the contention overhead. To improve decoding efficiency, Sampleless Wi-Fi leverages PHY hints to gradually screen out erroneous intermediate results during the decoding process.

We implement Sampleless Wi-Fi on the GNURadio/USRP platform. Evaluation results validate Sampleless Wi-Fi in reliably receiving and decoding legacy packets at sub-Nyquist rates across a wide range of channel conditions. Furthermore, from real traffic traces in a large-scale Wi-Fi network and across types of smartphone applications, we observe that Sampleless Wi-Fi largely improves Wi-Fi energy efficiency when compared to the state of the arts.

The contributions of this paper are summarized below.

- We provide a thoughtful study towards providing low-power Wi-Fi for future energy-constrained mobile devices. Our solution can be seamlessly integrated into existing Wi-Fi networks: it requires no PHY modifications on existing APs and completely conforms to 802.11 protocols.
- We propose a reliable reception pipeline that can decode legacy packets at sub-Nyquist rates under a wide range of SNRs. The key underlying technique is a new decoding algorithm based on time shift induced constellation maps.
- We prototype Sampleless Wi-Fi on USRP testbeds, and validate its performance with experiments and real traces.

The remainder of this paper is structured as follows. We begin in Section II with the design motivation. Section III gives an overview of the Sampleless Wi-Fi design. Section IV introduces time shift constellation mapping, which is the core technique of Sampleless Wi-Fi. Section V elaborates the detailed packet decoding process. System implementation and performance evaluation are presented in Section VI and Section VII, respectively. Section VIII discusses some practical issues when integrating Sampleless Wi-Fi with existing infrastructure. Section IX reviews related work, followed by the conclusion in Section X.

## II. Motivation

**The need for various sampling rates.** Today's Wi-Fi radios work at fixed clock rates to send and receive packets on preset channels. This fixed clock rate setting has worked well over the past decade, while recently becoming energy inefficient to support versatile mobile devices with various workloads. It has been reported that only a small fraction of devices sporadically make full use of high-speed Wi-Fi radios, while the majority normally under-utilize the channel capacity due to light workloads [4], [8]. Furthermore, the emerging Internet-of-Things (IoTs), including low-end smart devices and wearables, have increased the proportion of energy-constrained devices with light workloads.

**Reliability issues.** Recent years have witnessed advances in developing low-power Wi-Fi. The fundamental means is to turn off/down the radio clock rate, such as power saving mode (PSM) [9] or downclocking techniques [3]–[5]. Collectively, they have demonstrated the potential benefits of downclocking Wi-Fi radios, while they have limited use cases. PSM works only in limited scenarios as it causes additional latency and requires accurate traffic arrival estimation. Downclocking techniques outperform PSM in that they work regardless of traffic patterns. However, the downside is that they either require preamble modifications [3], sparsity in PHY that cannot be satisfied in today's Wi-Fi [4], or a significant margin between channel quality and data rate [5]. The main obstacle is that downclocking requires substantial redundancy, while today's PHY only leaves marginal redundancy by employing OFDM and rate adaptation. Therefore, these innovations cannot be directly realized in today's Wi-Fi networks, or only work under very limited scenarios.

**Observation.** The above situation evokes a similar picture in rateless codes, which allows transmitters to send packets with high modulation schemes that receivers cannot decode under low or medium SNR conditions. The wisdom of rateless codes is to gradually add redundancy in extra transmissions until the packets can be decoded. Inspired by rateless codes, we can add redundancy using retransmissions to assist the receivers to recover under-sampled packets. By scaling the numbers of transmissions, we can use flexible sampling rates under various channel conditions. As channel resources are frequently under-utilized by energy-constrained devices, extra

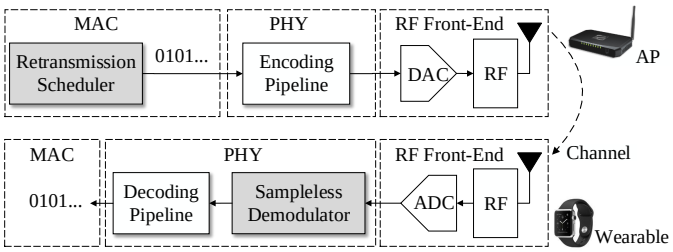



Fig. 2. Architecture of Sampleless Wi-Fi Design.

retransmissions are affordable. According to the measurements in [3], the energy used for retransmission packet reception is negligible compared to the energy saving of down-sampling.

## III. Sampleless Wi-Fi Overview

Sampleless Wi-Fi facilitates clients to reliably receive and decode legacy Wi-Fi packets at flexible sampling rates lower than the Nyquist criterion. It provides robust under-sampled reception under various SNR conditions. Consequently, low-end devices like wearables can downclock their sampling rates to save energy, while still enjoying comparable packet reception rates as traditional Wi-Fi receivers.

Architecturally, Sampleless Wi-Fi borrows from the wisdom of rateless codes in that an undecodable packet can be recovered by transforming original constellation symbols to points in a higher dimensional space through multiple transmissions of correlated symbols. Sampleless Wi-Fi recovers under-sampled baseband samples by multiple retransmissions. Different from rateless codes, Sampleless Wi-Fi requires no PHY modifications at the transmitters, but rather creates correlated symbols through time shifts at the receivers.

Fig. 2 shows the system flow of Sampleless Wi-Fi transmission and reception. Sampleless Wi-Fi extends legacy Wi-Fi by adding the following two core components to legacy protocol stack.

- **Retransmission Scheduler** residing atop AP's MAC. Sampleless Wi-Fi simply adds the retransmission scheduler at the MAC of Wi-Fi APs, while other MAC blocks such as the CSMA mechanism and the entire PHY still conforms to standard Wi-Fi. When an AP sends packets to a wearable device, the retransmission scheduler overrides the standard retransmission scheme and allows the AP to send multiple repeated frames in a single TXOP.
- **Sampleless Demodulator** incorporated in the client's decoding pipeline. To reliably receive and decode packets at sampling rates below the Nyquist frequency, the sampleless demodulator is incorporated at the PHY of wearables. A wearable receiver first feeds under-sampled baseband samples to the sampleless demodulator to recover the encoded bits, and then forwards the recovered bits to the traditional decoding pipeline, including deinterleaving, error correction codes decoding, descrambling, and cyclic redundancy check (CRC).

Sampleless Wi-Fi makes PHY changes (decoding pipeline algorithms) at the client side only, while the AP still conforms to the legacy PHY/MAC except that packet scheduling is performed by the retransmission scheduler, which can be easily implemented in commercial APs. Therefore, it allows next-generation wearables implemented with the new decoding pipeline to receive legacy packets from existing Wi-Fi APs at sub-Nyquist sampling rates with high reliability.

In what follows, we detail the essential components in Sampleless Wi-Fi.

## IV. Creating Diversity Through Multiple Retransmissions

The core and challenging part of Sampeless Wi-Fi is to design a receiver that can recover under-sampled constellation symbols through multiple retransmissions. In particular, the Sampleless Wi-Fi receiver must overcome the following challenges: (i) it must be able to receive and decode legacy packets without any modifications at the sender; (ii) it must be resilient to the changes of sampling rates and the variations of unpredictable channel conditions. To meet these challenges, we propose *time shift constellation mapping* to create diversity through multiple transmissions of repeated legacy frames.

### A. Frequency Aliasing Effect

When a receiver samples signals at sub-Nyquist criterion, the sample rate is insufficient to capture the changes in the signal, thereby causing frequency aliasing, that is, the signal spectrum folds up.

We illustrate the frequency aliasing effect on a typical OFDM symbol sampled at 1/2-Nyquist rate. The transmitted $N$ samples in one OFDM symbol can be expressed as $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{j2\pi kn}{N}} X_k, n = 0, ..., N-1$, where $X_k$ is the data encoded on the $k$th subcarrier. When $x[n]$ is sampled at 1/2-Nyquist rate, the receiver yields

$$\begin{aligned} y[n] &= x[2n] * h[2n] + w[2n] \\ &= \frac{1}{N}\sum_{k=0}^{N-1} e^{\frac{j4\pi kn}{N}} X_k H_k + w[2n], n = 0, ..., \frac{N}{2} - 1, \end{aligned} \quad (1)$$

where $H_k$ is the channel response at the $k$th subcarrier, and $w[n]$ is noise. Then, the receiver performs $\frac{N}{2}$-point FFT to obtain received subcarrier data $Y_{k'}, k' = 0, ..., N/2 - 1$.

$$\begin{aligned} Y_{k'} &= \sum_{n=0}^{\frac{N}{2}-1} e^{\frac{-j2\pi k'n}{N/2}} y[n] \\ &= \frac{1}{N}\sum_{n=0}^{N/2-1}\left(\sum_{k=0}^{N-1} e^{\frac{j2\pi(k-k')n}{N/2}} X_k H_k + w[2n] e^{\frac{-j2\pi k'n}{N/2}}\right) \\ &= \frac{X_{k'}H_{k'} + X_{k'+\frac{N}{2}} H_{k'+\frac{N}{2}}}{2} + \frac{1}{N}\sum_{n=0}^{\frac{N}{2}-1} w[2n] e^{\frac{-j2\pi k'n}{N/2}}. \end{aligned} \quad (2)$$

We observe that the receiver yields compound frequency samples: the $k$th sample after FFT consists of two subcarriers $X_{k'}$ and $X_{k'+N/2}$. Correspondingly, when sampling rate is downclocked to be $1/M$ Nyquist rate, each frequency sample captured at the receiver consists of $M$ compound samples.

Though several recent efforts [4], [5], [10] have employed sparse recovery techniques to decode symbols under frequency aliasing, they require certain sparsity properties that can hardly be guaranteed in general Wi-Fi transmissions. Current Wi-Fi transmissions have almost made the best use of channel resources, and leave little redundancy that can barely support direct decoding at sub-Nyquist rates. For example, Enfold [5] uses BPSK and requires very high SNR (>25 dB), while commodity APs normally employs much higher modulation schemes such as QAM-64 in high SNR conditions.

In order to decode under-sampled packets in various SNR conditions, in the next subsection we propose to exploit new diversity through multiple retransmissions. Specifically, we map under-sampled constellation symbols to higher dimensional space by utilizing the time shift effect.

### *B. Time Shift Induced Constellation Mapping*

Inspired by the design rationale of rateless codes, we use multiple retransmissions to map under-sampled constellation symbols to higher dimensional space. The essential intuition behind rateless codes is that the minimal distance between nearby constellation points can be increased when we map the conventional symbols to a higher dimensional space by sending a batch of correlated symbols. However, rateless codes cannot be implemented in commercial Wi-Fi APs due to PHY modifications, i.e., transmitting correlated packets instead of legacy packets. To overcome this obstacle, we exploit new diversity that can be obtained from the existing retransmission scheme in Wi-Fi networks.

**Exploiting the time shift effect.** Recall that frequency aliasing results in compound constellation symbols consisting of multiple frequency samples at different subcarriers. Intuitively, as the frequency of each subcarrier is unique, if we add frequency-specific operations before the decoding process, we can yield new compound constellation symbols at the receivers without modifying the transmitted packets. Specifically, we leverage the *phase-rotation property* [11] of the Fourier transform, that is, a shift in the time domain translates into a phase rotation in the frequency domain. Mathematically, after shifting the input signal $x[n]$ by $\tau$ samples, and performing $N$-point FFT, the input signal in the frequency domain changes by

$$\hat{X}_k = e^{\frac{j2\pi k\tau}{N}} X_k, \tag{3}$$

where $X_k$ and $\hat{X}_k$ are the signals in the frequency domain before and after the time shift, and $k$ is the subcarrier frequency. Thus, each subcarrier has a unique rotation speed with respect to time shift.

**Constellation diversity induced by time shifting.** Based on the above observation, if we inject different time shifts for repeated packets, we can obtain different constellation symbols, thereby creating multiple constellation maps at the receiver. Suppose we shift received signals by $\tau_i$ for the $i$th retransmission. According to Eq. (2) and (3), the $k$th subcarrier data sampled in the $i$th retransmission (noise omitted) is

$$Y^i_{k'} = \frac{1}{2}\left(e^{\frac{j2\pi k'\tau_i}{N}} H^i_{k'} \cdot X_{k'} + e^{\frac{j2\pi(k'+\frac{N}{2})\tau_i}{N}} H^i_{k'+\frac{N}{2}} \cdot X_{k'+\frac{N}{2}}\right), \tag{4}$$

Note that the subcarrier data is the sum of rotated versions of $H^i_{k'} \cdot X_{k'}$ and $H^i_{k'+\frac{N}{2}} \cdot X_{k'+\frac{N}{2}}$, whose rotations are different and controlled by $\tau_i$. We can rewrite each component of $Y^i_{k'}$ in the form of phase and amplitude as

$$Y^i_{k'} = \frac{1}{2}\left(\left|H^i_{k'} X_{k'}\right| e^{j2\pi k'\tau_i/N + \varphi(H^i_{k'} X_{k'})} + \left|H^i_{k'+\frac{N}{2}} X_{k'+\frac{N}{2}}\right| e^{j2\pi(k'+\frac{N}{2})\tau_i/N + \varphi\left(H^i_{k'+\frac{N}{2}} X_{k'+\frac{N}{2}}\right)}\right), \tag{5}$$

where $\varphi(\cdot)$ and $|\cdot|$ are the phase and amplitude of a complex constellation point, respectively. Due to the different rotation speed of each component, the compound symbol $Y^i_{k'}$ has a unique phase and amplitude corresponding to the time shift $\tau_i$. Consequently, we generate a distinct constellation map for each retransmission by adding different time shifts.

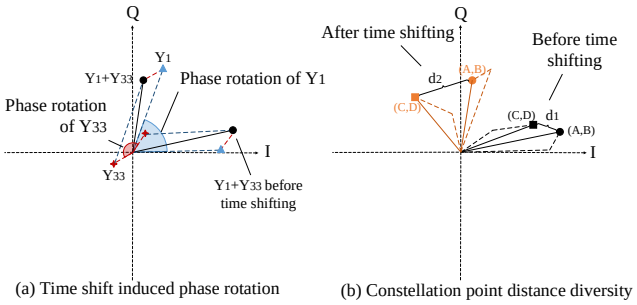
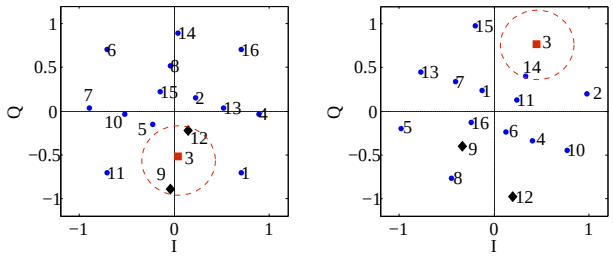



Fig. 3. Illustration of time shift effect.

Fig. 3 illustrates how a time shift at the receiver with 1/2-Nyquist rate generates a new constellation map. We consider a compound constellation sample consisting of received frequency samples $Y_1 = H_1 X_1$ and $Y_{33} = H_{33} X_{33}$, where $H_k$, $X_k$ denote the channel response and transmitted data at the subcarrier $k$ ($k = 1$ or 33), respectively. As illustrated in Fig. 3(a), after a time shift at the receiver, $Y_{33}$ rotates a larger angle than $Y_1$ according to Eq. (3). As a consequence, both the amplitude and phase of the compound symbol $Y_1 + Y_{33}$ change after time shifting. Such changes are reflected in the constellation map, as shown in Fig. 3(b). Without loss of generality, we consider four constellation points, denoted as $A, B, C, D$ (e.g., $A$ can be "1101" in 16QAM). Due to frequency aliasing, the receiver only obtains compound points, which are irregularly scattered in the constellation map according to Eq. (2). Suppose the compound points $(A, B)$ and $(C, D)$ have a very small distance $d_1$ in the constellation map, which can be very hard to distinguish at low SNRs. After time shifting as shown in Fig. 3(a), the compound points $(A, B)$ and $(C, D)$ change in both amplitudes and phases. As components in $(A, B)$ and $(C, D)$ have different amplitudes, the amplitudes and phases changes of $(A, B)$ and $(C, D)$ are also different Consequently, the distance between $(A, B)$




(a) Constellation map without time shift. (b) Constellation map with time shift.

Fig. 4. Constellation diversity induced by time shift. Constellation maps are constructed based on packets received at 1/4-Nyquist rate using USRP.

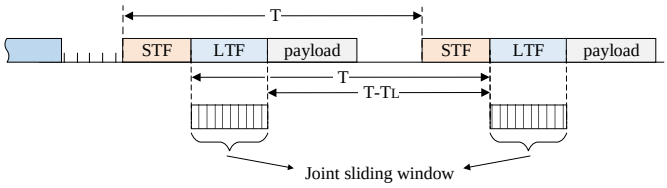



Fig. 5. Joint synchronization over consecutive frames.

and $(C, D)$ after time shifting, denoted as $d_2$, also changes. Generally, neighboring points in the constellation map become faraway points, and vice versa. As such, a time shift creates a new constellation map with different point distances.

By combining multiple transmissions, we can map constellation points to higher dimensional space with a larger minimum distance. Fig. 4 shows the constellation maps of BPSK-modulated packets captured by USRP at 1/4-Nyquist rate. Note that frequency aliasing at 1/4-Nyquist rate converts the BPSK constellation to a 16-point irregular constellation. The red circle denotes the minimum distance required to distinguish two points in demodulation. In Fig. 4(a), point 3 is likely to be mixed up with its neighboring points 9 and 12 during demodulation due to channel noise. In another transmission with time shifting (Fig. 4(b)), constellation points are completely shuffled: points 9 and 12 move far beyond the minimum distance of point 3. This distance diversity benefits demodulation by separating mixed-up points. By combining these two transmissions, the minimum distance in one transmission (e.g., the distance between points 9 and 3) is transformed to a higher dimensional metric (e.g., accumulated distance in two transmissions), which is generally much larger and thereby more resistant to noise.

## V. Sampleless Decoding

Thus far we have explained how a sub-Nyquist receiver can create distinct constellation maps through time shifting without any modifications at the sender's side. Based on this basic idea, we design a Sampleless Wi-Fi receiver that harvests the diversity in retransmissions to receive and decode under-sampled frames.

### A. Joint Synchronization

A Wi-Fi receiver continuously performs idle listening in order to detect Wi-Fi preambles and synchronize arriving

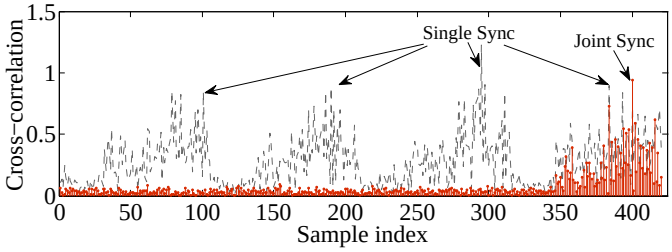



Fig. 6. Cross-correlation responses at 1/4 Nyquist rate. We set up a USRP link to capture raw data samples of four consecutive Wi-Fi packets at 10 dB SNR. Joint synchronization combines four LTFs.

frames. The Wi-Fi preamble consists of Short Training Field (STF) and Long Training Field (LTF), which are used for synchronization and channel estimation, respectively. The legacy STF contains ten repetitions of a 16-sample sequence, while the LTF contains two repetitions of a 64-sample symbol. Wi-Fi receivers generally leverage the auto/cross-correlation properties of the STF to synchronize incoming frames.

As the number of the STF is reduced under sub-Nyquist sampling, the correlation results of the STF are not strong enough for fine synchronization. Experiments in [5] have shown that the cross-correlation result of the under-sampled LTF can be used for synchronization in a high SNR range (>13 dB). However, when the SNR is poor, noise cannot be canceled out during cross-correlation.

To perform synchronization for sub-Nyquist receivers under low SNR conditions, we employ a joint synchronization algorithm that utilizes multiple preambles in consecutive transmissions to add LTF samples in cross-correlation. It is worth noting that commercial APs can precisely control inter-frame timing with quite a small clock error (e.g., 40 MHz clock oscillators) that is much lower than the frame synchronization error (e.g., due to multipath delays). Thus, the AP can send repeated frames at a fixed interval, e.g., SIFS in one TXOP, which is predetermined and known by the receiver. As such, the receiver can perform a joint cross-correlation across the LTFs of the consecutive frames. Fig. 5 illustrates the transmission of two repeated frames. First, the receiver uses the STF and LTF to synchronize the first frame, and buffers the baseband samples. After detecting the second frame, the receiver computes one joint cross-correlation over the STFs of the two frames. The joint cross-correlation can be realized by modifying the sliding window. We denote the length of the LTF as $T_L$ and the time offset between two consecutive frames as $T$. Specifically, instead of using a sliding window of length $T_L$, a joint sliding window consisting of two windows of length $T_L$ spaced by $T-T_L$, as shown in Fig. 5. Consequently, the samples for cross-correlation is doubled, thereby exhibiting stronger cross-correlation properties that are more robust under poor SNR conditions. Then, we synchronize each frame based on the correlation spike and the corresponding time offset $T$.

Fig. 6 illustrates the merits of joint synchronization at 1/4 Nyquist rate. As the LTF samples are reduced by quarter at 1/4 Nyquist rate, the cross-correlation result of a single LTF contains multiple comparable peaks at a low SNR (10 dB), and thus the receiver can easily aligned to a wrong peak. By

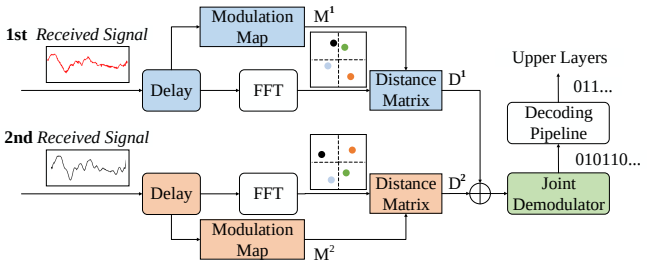



Fig. 7. Sampleless Demodulator.

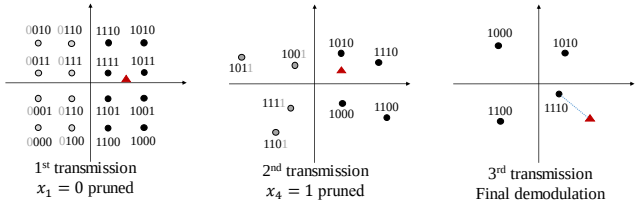



Fig. 8. An example of constellation pruning using NCL. The triangle point is the received symbol of 1110. Grey points are discarded as they contain bits with low NCLs.

jointly utilizing four LTFs, we can achieve a desirable cross-correlation result where the highest peak corresponds to the full alignment of OFDM symbols.

### *B. Decoding Over Multiple Retransmissions*

Prior to data decoding, the receiver performs channel estimation based on LTF. The receiver estimates channel response of each transmission independently, and thus the packet reception is not compromised by the variance of channel response across transmissions. The only departure from the standard channel estimation is that Sampleless receivers obtain compound frequency responses in the form of (2) due to frequency aliasing [5]. Then, the receivers construct a constellation map based on channel estimation, and combine the frequency responses in multiple retransmissions to convert compound constellation symbols to a higher dimensional space to recover the original symbols. The high-level pipeline is illustrated in Fig. 7.

The Sampleless Demodulator combines multiple packets based on the maximum likelihood (ML) algorithm. From $L$ transmissions, the receiver obtains $L$ symbols $Y = y_1, ..., y_L$ of a symbol $x_j$. An ideal ML decoder minimizes the probability of error with respect to a uniform prior, which is,

$$\hat{X} = \arg\min_{x_j} \sum_{i=1}^{L} \frac{1}{\sigma^2} \|\bar{y_i} - \tilde{x_j}\|^2, \tag{6}$$

where $\hat{X}$ is the decoded symbol, $\|\bar{\|}^2$ the squared Euclidean distance in the constellation map, and $\sigma$ the noise variance. With more transmissions, the distance in the constellation diagram increases and it becomes easier to separate symbol $x_j$ from neighboring constellation points.

The ideal ML decoder can achieve optimal capacity over AWGN channels. However, with multiple transmissions, the decoding complexity grows dramatically. Decoding computations consume power and computing resources, which are limited at low-end devices. A more efficient decoding algorithm would extend Sampleless Wi-Fi to more resource-constrained devices, or allow a device to decode more retransmissions for more energy savings. We observe that there are unnecessary computations when the ML decoder is extended to multiple transmissions. To reduce the computational complexity, we continuously prune the constellation based on *soft hints*, which are the outputs from the standard ML or maximum a posteriori probability (MAP) decoder. Soft hints represent the reliability of the decoding bits, e.g., posteriori Log-Likelihood Ratios (LLRs) [12]. Upon receiving a symbol $x$ , the LLR of its $k$th bit $x_k$ is

$$LLR(x_k) = \log \frac{P(x_k = 1|r)}{P(x_k = 0|r)}. \tag{7}$$

The basic idea of soft hint based constellation pruning is that instead of searching all possible constellation points, we prune the constellation by discarding the unreliable points based on their LLRs, thereby reducing the search space and hence complexity. However, due to channel impairments, the LLR itself may become unreliable and cannot represent the real confidence level of a bit. Therefore, we adopt normalized confidence level (NCL) as the pruning criteria,

$$NCL(x_k) = \frac{|LLR(x_k)|}{\max(|LLR(x_k)|)}. \tag{8}$$

Suppose each pruning screens out $pN$ constellation points out of total $N$ constellation points, and the total computations to decode one transmission is $C_N$. In the $i$th transmission, we only require $(1-p)^{i-1}C_N$ computations. Fig. 8 illustrates the constellation pruning using NCL. As $x_1 = 1$ in the first transmission cannot satisfy the NCL criteria, we discard the eight corresponding constellation points, and leave the reminders to compute the accumulated distance matrix upon the next transmission. Similarly, the points with $x_4 = 1$ are discarded in the second transmission according to the NCL criteria. After receiving three transmissions, only four points are left for final demodulation, which reduces unnecessary computations. Compared to the standard ML decoder, we save $50\%$ computations in decoding the second transmission, and save $75\%$ in decoding the third transmission.

The detailed decoding process is summarized in Algorithm 1. For each packet received during the TXOP duration, the receiver applies a delay to the received signals and obtains time-shifted compounded constellation symbols. Then, the receiver generates a modulation map based on the amount of time shift. The receiver computes the squared distance matrix $D^2$, which consists of the square Euclidean distance of each compounded constellation symbol to each constellation symbol in the modulation map. The receiver updates the squared distance matrix $D^2$ by accumulating the corresponding squared distances of all transmissions. Then, the receiver screens out constellation symbols based on the NCL criteria, and shrinks the distance matrix accordingly. As such, the distance computations are gradually reduced. Finally, the receiver demodulates the packet based on the updated $D^2$

in the final transmission, and sends an acknowledgment frame (ACK) if demodulated bits pass the checksum.

**Algorithm 1:** Sampleless Decoding

**Data**: Compounded constellation symbols
**while** *A new packet received in the TXOP duration* **do**
  Generate time shift induced modulation map;
  Compute squared distance matrix $D^2$;
  Update squared accumulated distance matrix $D_a^2 \leftarrow D_a^2 + D^2$;
  Prune bits based on $NCL$;
**end**
**if** *Decoding is successfull* **then**
  send ACK;
**else**
  wait;
**end**

## VI. Implementation

We implement Sampleless Wi-Fi atop the OFDM implementation of the GNURadio/USRP platform. We adopt the legacy PHY layer convergence procedure (PLCP) format of 802.11n, where the PLCP preamble consists of 2 OFDM symbols for the short training field (STF) and 2 OFDM symbols for the LTF. Nodes in our experiments are USRP N210 devices equipped with RFX2450 daughterboards as RF frontend, which operates in the 5.1-5.2GHz range. Each USRP node is connected to a DELL OptiPlex 9010 desktop with Intel i3 dual-core CPU running Ubuntu 10.04.

Since USRP cannot be downclocked while receiving full bandwidth signals, we digitally downscale the sampling rate after ADC by decimation. Specifically, we decimate every other sample for $50\%$ and three of every four samples for $25\%$. Our current implementation on USRP realizes delay lines by injecting time shifts digitally. Note that in a complete prototype or product, delay lines are implemented in the analog front-end between the ADC and the antenna. Different delays can be achieved using wires of different lengths [10]. The delay can be either an integer number or fraction number of the ADC sample interval, while the only constraint is that the delay cannot be an integer number of the OFDM symbol interval.

We modify the synchronization algorithm and decoding pipeline in the USRP Hardware Drive (UHD) to empower sampleless reception, as specified in Section V. Due to large processing delay of USRP and its connected computer, CSMA MAC cannot be performed in real-time. To evaluate energy consumption of Sampleless Wi-Fi, we leverage real-world Wi-Fi traces and emulate the MAC protocol. The multiple retransmissions of Sampleless Wi-Fi are realized using the 802.11 TXOP scheme [7], in which packets within one TXOP duration are sent back-to-back without requiring additional channel contention.

## VII. Evaluation

### A. Packet Reception Performance

A primitive function of Sampleless Wi-Fi is providing reliable packet reception at sub-Nyquist rates under various SNR conditions. We use USRP testbeds to evaluate its synchronization and decoding performance.

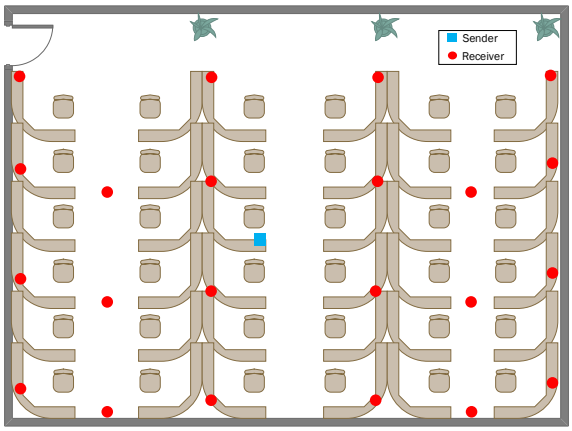


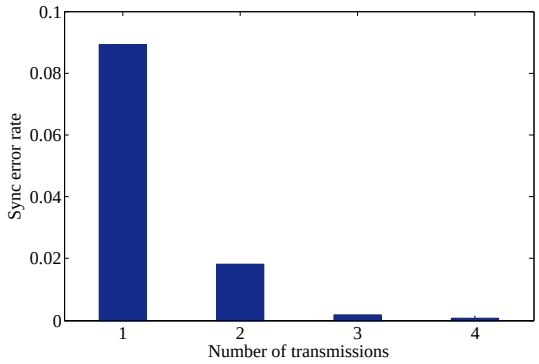

Fig. 9. Testbed layout. The sender is placed in the center of the room, and the receiver is placed in different locations scattered in the room.

**Experimental Setup.** We conduct our experiments using two USRP nodes in a 10 m $\times$ 10 m office. We vary the location and transmission power of USRP nodes to test Sampleless Wi-Fi under different channel conditions. We fix the location of the sender node and move the receiver node to different locations in the office, as shown in Fig. 9. Unless otherwise stated, Sampleless Wi-Fi adds one (three) retransmissions for each packet at 1/2 (1/4)-Nyquist rate.

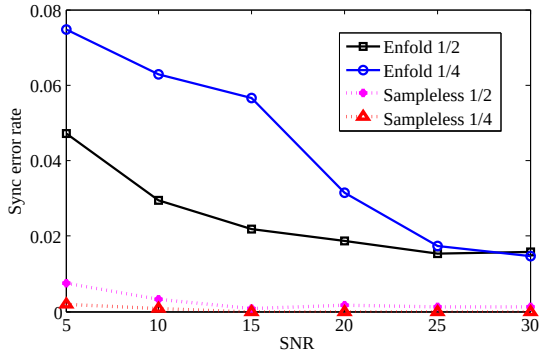
We compare Sampleless Wi-Fi with the state-of-the-art downclocking technique Enfold [5], which exploits under-sampled constellation characteristics for decoding. The comparison shows the merits of combining time shifted constellations in Sampleless Wi-Fi. We also show the performance of the standard 802.11 packet reception at the Nyquist rate as the benchmark.

**Synchronization.** First, we show that the Sampleless Wi-Fi addresses the synchronization issue at sub-Nyquist rates under low SNRs. We compare the joint synchronization over multiple transmissions with the previous synchronization algorithm in Enfold [5], which uses the cross-correlation of LTF in one transmission. We set the USRP receiver at full sampling rate to capture raw samples, and perform different synchronization algorithms offline for comparison. We use the synchronization algorithm at full sampling rate as the benchmark, and estimate the relative error of synchronization at sub-Nyquist rates. To perform under-sampling offline, we digitally downscale the sampling rate by decimation. We continuously send 5000 packets under each setting to estimate the error rate.

Fig. 10(a) evaluates the performance of joint synchronization at 1/4-Nyquist rate with various numbers of transmissions. The synchronization error drops significantly as the number of transmissions increases, and approaches zero at four transmissions. This shows the merits of using multiple LTFs to compensate the lack of LTF samples in one transmission. The number of samples at four transmissions equal to the samples at full rate in one transmission, and thus synchronization in these two cases performs the same.




(a) Joint synchronization error rate with various numbers of transmissions. SNR is set to be 10 dB.




(b) Comparison with Enfold under various SNRs.

Fig. 10. Synchronization performance evaluation.

The observation in Fig. 10(a) explains the results in Fig. 10(b). For Enfold, the synchronization error rate grows substantially when SNR drops below 20 dB. This is because the lack of samples at sub-Nyquist rates fails to show strong correlation property over noise. Sampleless Wi-Fi employs joint synchronization to amortize this issue by combining multiple transmissions.

**Retransmission diversity.** The decoding performance is dictated by the minimum distance between any two constellation points [13]. We set this experiment as follows. The USRP transmitter sends 5000 QPSK-modulated packets to the receiver node at 1/4-Nyquist rate. The receiver estimate channels and then derives the compound constellation map after channel equalization.

Fig. 11 compares the minimum distances by combining multiple transmissions. The minimum distance is normalized with respect to that of full clock rate. We observe that the minimum distance in one transmission is quite small. This is because frequency aliasing results in an irregular constellation of compound symbols, some of which are co-located, as illustrated in Fig. 4. This is the reason Enfold requires high SNR for decoding. By adding a time shift into each received packet, Sampleless Wi-Fi achieves $\times 1.5$-$\times 2$ minimum distance compared to standard retransmissions. As such, Sampleless Wi-Fi can reliably decode under-sampled packets under low SNR conditions.

**Sampleless demodulation.** We evaluate the reliability of Sampleless Wi-Fi under various SNR conditions. We use the same sampling rate for Enfold and Sampleless Wi-Fi for comparison, while using the full sampling rate for 802.11 as the benchmark. To keep reasonable decoding overhead, we use 1/2-Nyquist rate for PSK and 1/4-Nyquist rate for QAM by default.

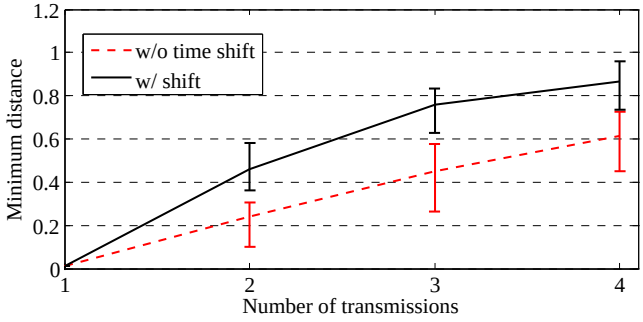
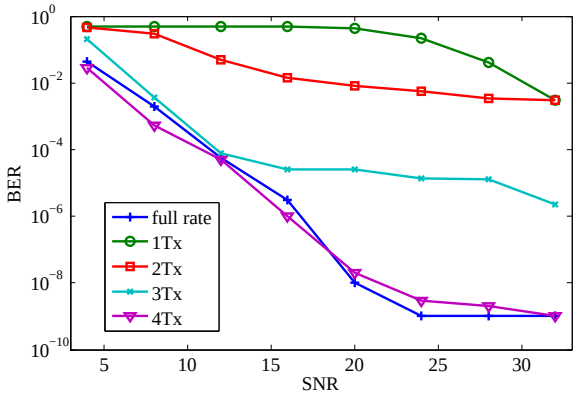
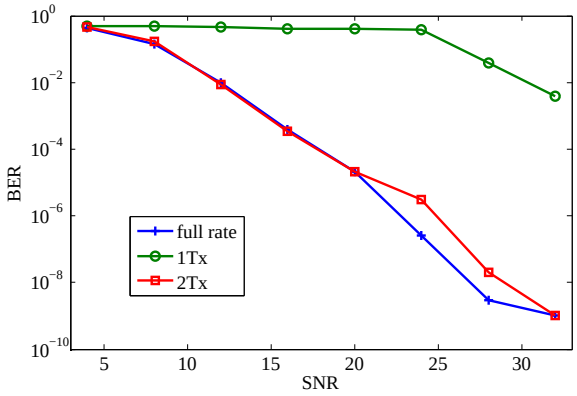



Fig. 11. Minimum distance at 1/4 Nyquist rate.



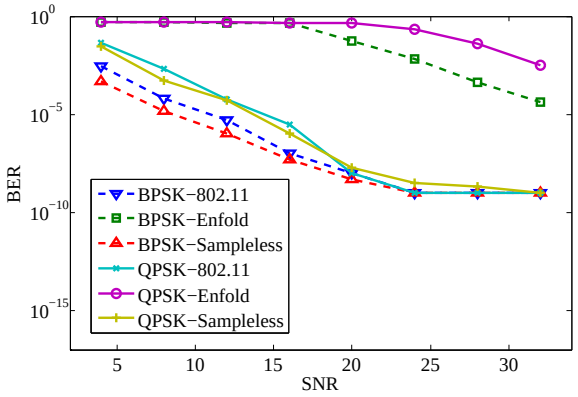

(a) QPSK at 1/4-Nyquist rate



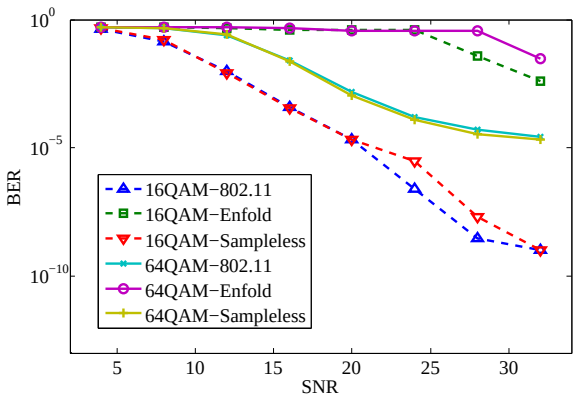

(b) 16QAM at 1/2-Nyquist rate

Fig. 12. BER under various numbers of transmissions.

Fig. 12 shows the BER of Sampleless Wi-Fi with various numbers of transmissions. We have the following observations: i) the down-sampled receiver suffers significantly high BERs for SNRs below 25 dB, and ii) the receiver with $1/M$-Nyquist rate requires $M$ transmissions to achieve comparable BERs with respect to the receiver with the full sampling rate. The results demonstrate the necessity and merits of leveraging retransmissions for reliable under-sampled packet decoding.

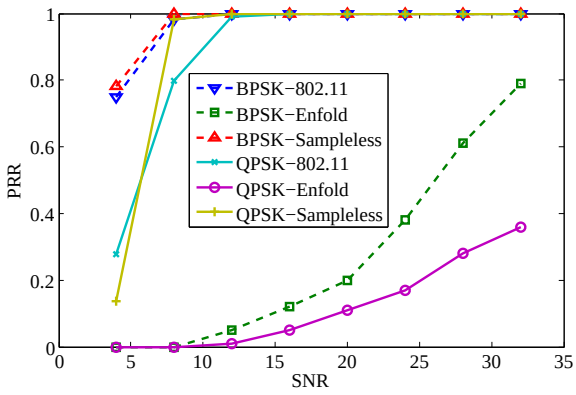



(a) BPSK and QPSK

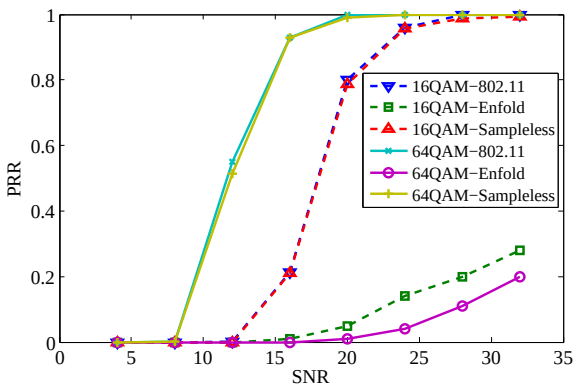



(b) 16QAM and 64QAM

Fig. 13. BER under various SNR conditions.




(a) BPSK and QPSK




(b) 16QAM and 64QAM

Fig. 14. PRR under various SNR conditions.

We further compare the BER of Sampleless Wi-Fi and Enfold under various SNR conditions in Fig. 13. Sampleless Wi-Fi achieves significantly lower BER than Enfold across all modulations and SNRs. This is because Enfold directly decodes packets based on one constellation map, while frequency aliasing results in some co-located constellation symbols that are easily mixed-up due to noise, as illustrated in Fig. 4. Sampleless Wi-Fi achieves comparable BER compared to 802.11, which demonstrates that the down-sampling's negative effect on decoding is completely compensated by extra retransmissions.

**Overall packet reception.** We now evaluate the overall packet reception performance under a range of SNR conditions. We first measure the packet reception ratio (PRR), which is defined to be the ratio of the number of correctly decoded packets to the total number of packets. We use the same sampling rate for Enfold and Sampleless Wi-Fi, while using full sampling rate for the standard 802.11. The packet length is set to 100 bytes, and the coding rate is set to 1/2 for all modulations.

Fig. 14 show the results of these experiments. As shown in Fig. 14(a), Sampleless Wi-Fi achieves almost 100% PRR when SNR is over 8dB and 12dB for 1/2 BPSK and 1/2 QPSK, which is consistent with the performance of the standard 802.11. Fig. 14(b) also shows that Sampleless Wi-Fi and the standard 802.11 achieve comparable PRR for 1/2 16QAM and 1/2 64QAM. The results demonstrate that Sampleless Wi-Fi can maintain the decoding performance of the standard 802.11 for different modulation schemes under various SNR conditions. On the other hand, Enfold suffers from low PRR when SNR is lower than 20dB. This is because the Enfold needs under-selected modulation schemes, i.e., low-order modulation scheme under high SNR conditions, to create enough redundancy in PHY for decoding.

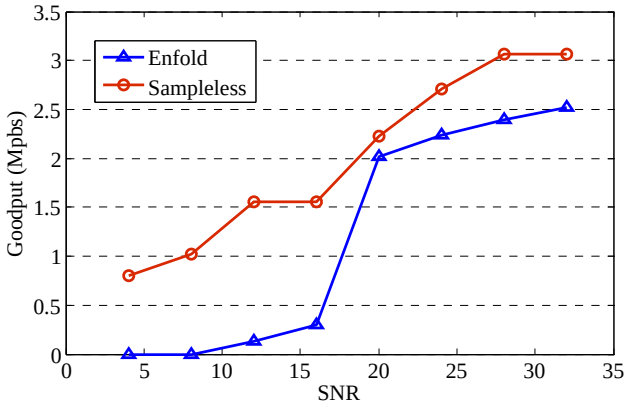



Fig. 15. Goodput under various SNR conditions. MCSs are selected according to the SNR-based rate adaptation.

As Wi-Fi transmissions normally select different modulation and coding schemes using rate adaptation schemes, we further

compare the goodput of Sampleless Wi-Fi and Enfold with rate adaptation enabled. The data rate for each transmission is selected based on the link's SNR according to the standard SNR-data rate mapping table as listed in [14]. We observe that Sampleless Wi-Fi achieves higher goodput compared to Enfold under all SNR conditions. When SNR is lower than 20dB, the goodput of Enfold diminishes quickly, while Sampleless Wi-Fi still maintains reasonable goodput. It shows that Sampleless Wi-Fi is applicable to receive legacy packets with rate adaptation enabled under various SNR conditions.

### B. Energy Efficiency Performance

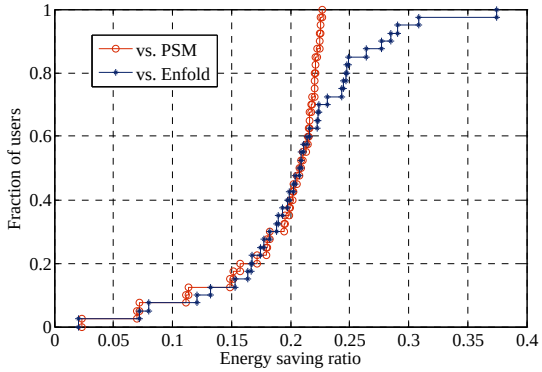
Since we cannot directly measure power consumption on Wi-Fi devices with sub-Nyquist rates, we turn to trace-driven simulations to evaluate the energy efficiency performance.

*1) Simulation Methodology:* We adopt a Wi-Fi power model [5], [15] to parameterize the network energy consumption of each device given its traffic traces. A Wi-Fi radio stays in a high power state when it is actively sending or receiving packets, while it switches to the idle state once packet transmitting/receiving completes. The radio further enters the light sleep state and then the deep sleep state if there is no network activities for a while. Detailed parameters are given in [4], [15]. In our simulation, we use the power model to infer the instantaneous power of Wi-Fi radio, and combine traffic traces to compute the overall energy consumption. As Wi-Fi traces do not contain SNR, we estimate the SNR of each packet based on the data rate according to [14].

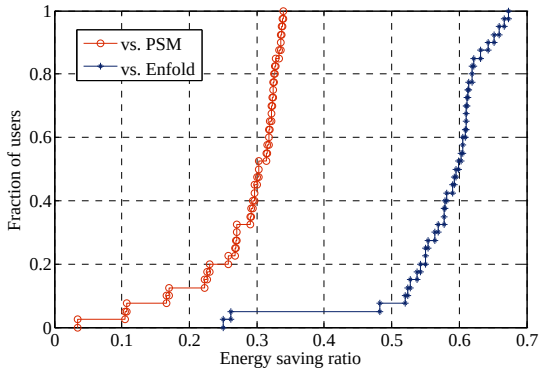
**Large-scale Wi-Fi network traces.** We use the SIGCOMM'08 traces [16] to evaluate our design in real large-scale Wi-Fi networks. The SIGCOMM'08 trace captures Wi-Fi traces during the conference event that has a peak of 31 clients. The average inter-arrival times are 47 ms and 88 ms for TCP and UDP traffic, respectively.

**Smartphone app traces.** To further understand the detailed energy efficiency of Sampleless Wi-Fi under various types of applications, we collect Wi-Fi traces of popular smartphone apps of different types. We cover both delay sensitive and non-sensitive apps, low data rate apps and bandwidth-hungry apps. Specifically, we select *Skype* (VoIP, 118 kbps), *FaceTime* (video call, 762 kbps), *Chrome* (website browsing, 196 kbps), *YouTube* (video streaming, 2150 kbps), *Instagram* (online picture browsing, 877 kbps), *Spotify* (music streaming, 636 kbps). These app traces are collected as follows. We set up a sniffer (ThinkPad X201) using *Wireshark*, and place it near an iPhone 5S, which is associated with an 802.11n AP. Each app is actively used for two minutes. There are around ten other active Wi-Fi devices on the same channel.

**Baselines.** We compare Sampleless Wi-Fi with the standard power saving mode (PSM) in 802.11 [9] and the state-of-the-art downclocking approach Enfold [5]. As Enfold only experiences extremely high packet loss rate for higher modulation schemes, we fix the data rate of Enfold to be 6 Mbps (1/2 BPSK) as suggested in [5]. Note that the packet loss rates of Enfold are consistent with [5]. We set the same sampling rate for Enfold and Sampleless Wi-Fi for comparison.




(a) 1/2 Nyquist rate

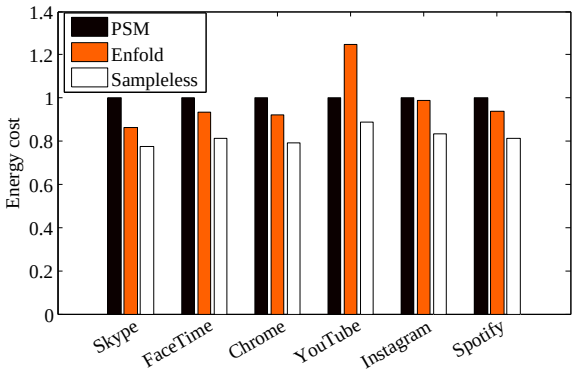



(b) 1/4 Nyquist rate

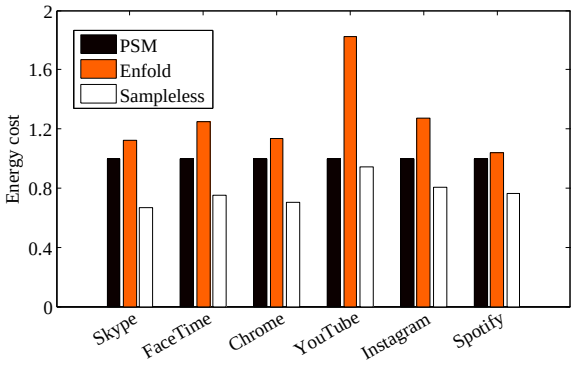
Fig. 16. Energy saving gain for large-scale networks.

*2) Results:* Fig. 16 shows the energy saving ratios of Sampleless Wi-Fi compared to PSM and Enfold. At 1/2 Nyquist rate in Fig. 16(a), for $80\%$ of the clients, Sampleless Wi-Fi saves $17.2\%$ and $16.7\%$ or more energy compared to PSM and Enfold, respectively. Though Enfold sets the same sub-Nyquist sampling rate as Sampleless Wi-Fi, it's packet loss rate is high under normal indoor SNR conditions, thereby incurring substantial retransmission overhead and extra packet decoding energy costs. Compared with PSM, Sampleless Wi-Fi achieves comparable BER under all SNR conditions, and still harvests the benefits of under-sampling. At 1/4 Nyquist rate in Fig. 16(b), the energy saving ratios of Sampleless Wi-Fi grows to $25.8/\%$ and $55\%$ compared to PSM and Enfold, respectively. This is because the packet loss rate of Enfold increases significantly, which overwhelms the benefits of lower sampling rate. Sampleless Wi-Fi overcomes this limitations by adding back-to-back retransmissions, which largely reduces the power consumption in all states at the cost of merely small duration increment in packet receiving state.

Fig. 17 compares the energy consumption under various types of smartphone apps. Sampleless Wi-Fi outperforms PSM and Enfold in all cases demonstrated. Sampleless Wi-Fi consumes $77.6\% - 88.7\%$ energy at 1/2 Nyquist rate, and $66.8\% - 94\%$ energy at 1/4 Nyquist rate compared to PSM. Sampleless Wi-Fi saves the most energy for Skype ($22.4\%$ and $33.2\%$), and the least energy for YouTube ($11.3\%$ and $6\%$). Skype provides a VoIP service that generates an ON/OFF UDP stream with intermittent short packets. The intermittent




(a) 1/2 Nyquist rate




(b) 1/4 Nyquist rate

Fig. 17. Energy saving comparison for various smartphone apps. The energy cost is normalized with respect to PSM.

short VoIP packets leave room for multiple retransmissions in Sampleless Wi-Fi, while the frequent packet arrivals prevent PSM from entering the sleep state. On the other hand, YouTube prefetches chunks of video data at high data rates every few seconds. Sampleless Wi-Fi yields little benefit due to the heavy workloads of YouTube, while PSM enters the sleep state after fast downloading periods. Sampleless Wi-Fi achieves appreciable energy savings for other apps: Chrome (20.7% and 29.6%), FaceTime (19% and 24.8%), Spotify (19.5% and 23.9%), and Instagram (16.8% and 19.6%). Enfold outperforms PSM at 1/2 Nyquist rate except for YouTube traces, while consuming more at 1/4 Nyquist. The main reason that cancels its downclocking benefit is due to packet loss at various SNRs (10-30 dB) in indoor environments.

## VIII. Discussion

Some practical issues are discussed in this section.

### A. Integration with Existing Wi-Fi Infrastructure

Sampleless Wi-Fi is a standard-compliant design that can be integrated with existing Wi-Fi networks. Sampleless receivers can communicate with commercial Wi-Fi APs and conform to 802.11 standards, as described below.

**AP modification.** To support Sampleless Wi-Fi receivers, APs need to add a new scheduling algorithm, i.e., the Retransmission Scheduler, at MAC as shown in Fig. 2. When an AP sends packets to a Sampleless Wi-Fi receiver, the Retransmission Scheduler overrides legacy retransmission algorithm. Instead of waiting for ACK, the Retransmission Scheduler proactively send one or multiple retransmission packets back-to-back in one TXOP period, as illustrated in Fig. 18. The Retransmission Scheduler is an add-on at MAC, and can be implemented on commodity devices, e.g., by patching `Openwrt` and network interface card (NIC) drivers.

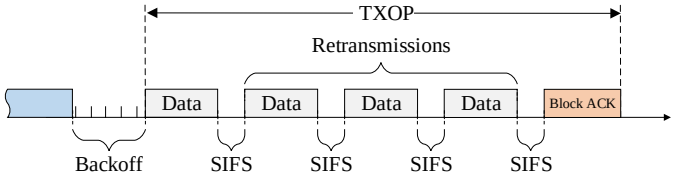



Fig. 18. Multiple retransmissions in one TXOP.

**Data scrambling.** In IEEE 802.11, the raw bits are scrambled before mapping to subcarriers. Scrambling is used to randomize the raw bits to avoid long strings of binary 1s or 0s. Our current implementation uses IEEE 802.11a scrambler, where the raw bits are XORed with the scrambler bits with an initial state. We set identical initial state for sampleless retransmissions, which is consistent to [17]. The receiver deterministically derives the scrambling sequence, and then performs descrambling before demodulation. As the initial state and the scrambling sequence are identical for retransmissions, the data encoded on a subcarrier would be the same for retransmissions.

**Rate adaptation.** Commercial APs may employ rate adaptation schemes to adjust the MCS of each packet. Commercially-implemented rate control algorithms, such as Auto Rate Fallback (ARF) [18] and MadWiFi Ministrel [19], mainly adjust the modulation and coding scheme of a packet by counting ACKs of previous transmission attempts. In Sampleless Wi-Fi, only one ACK is expected for a packet transmission and its following sampleless retransmissions. These retransmissions are considered as one transmission attempt, and thus are not counted as legacy retransmissions in rate adaptation schemes. Therefore, the MCS in sampleless retransmissions remain identical.

**Channel access.** Sampleless Wi-Fi is fully compatible with 802.11 protocols. Sampleless Wi-Fi nodes conform to CSMA/CA to contend channel, and enable multiple retransmissions of a packet using TXOP [7], which is proposed to improve MAC efficiency. During a TXOP period, the sender can send multiple packets without entering the backoff procedure, and then the receiver replies one block ACK. We leverage TXOP to enable consecutive retransmissions of repeated packets. As we target at devices with light workloads (e.g., VoIP, web browsing), the negative impact of increased transmission duration would be tolerable. It is worthwhile noting that the retransmissions in one TXOP are sent without contending channel, thereby incurring smaller overhead compared to legacy retransmissions. Our evaluation using SIGCOMM'08 trace [16] shows that extra retransmissions in Sampleless Wi-Fi do not cause congestion.

### B. Uplink Transmission

Sampleless Wi-Fi focuses on downlink transmission from APs to sampleless receivers. Due to the dominance of downlink traffic, the fraction of time for a client being in the transmitter mode is quite small, e.g., less than $2\%$ in SIGCOMM'08 trace [16], and the energy spent in the transmitter mode is negligible. Therefore, clients can switch to the full sampling rate for uplink transmission. As such, APs receive legacy uplink packets without any modifications. Note that the switching time between the full sampling rate and lower sampling rates is only several microseconds in state-of-the-art Wi-Fi chipsets, e.g., 9.5 $\mu s$ in MAXIM 2831 [3]. Sampleless Wi-Fi nodes can switch between clock rates within an SIFS interval, i.e., 16 $\mu s$ in IEEE 802.11a/n/ac. As the switching delay is rather low, a node can switch to full sampling rate right before it performs carrier sense for uplink transmission.

It is worth noting that Sampleless Wi-Fi still conforms to legacy carrier sense. In the downlink, the AP performs carrier sense and transmit packets to Sampleless receivers. As the AP operates at the full clock rate, it uses legacy criteria for carrier sense. In the uplink, Sampleless nodes turn to full clock rate for packet transmission, which also conforms to legacy Wi-Fi carrier sense and back off.

### C. Decoding Overhead

Compared to the standard Wi-Fi baseband, the decoding chain of Sampleless Wi-Fi introduces extra computations. These computations can be realized by employing standard blocks, .e.g, LLR, minimum distance decoding, correlation. It has been shown [5], [20] that power consumption incurred by extra computations in standard blocks are negligible compared to communication cost.

### D. Sampling Rate Adjustment

As reducing sampling rates incur extra retransmissions, there is a tradeoff between energy saving and goodput. As illustrated in Fig. 1, our vision is that heavy load nodes adopt normal sampling rate, while only medium and light load nodes adopt sub-Nyquist rates to save energy. Hence, Sampleless Wi-Fi mainly works for energy-constrained, light load devices, such as IoTs, wearables, and even smartphones with light load services.

Given the constraints and the valuations of energy saving and goodput, the AP can identify the optimal sampling rate for each node based on current and future traffic conditions. However, it is non-trivial to obtain the current and future traffic conditions of all nearby nodes. A practical way is to borrow contention window backoff algorithms or congestion control algorithms. In particular, we can employ a feedback control algorithm that adjusts sampling rate based on the network congestion conditions observed at the AP. If the collision probability is above a threshold, the AP "backs off" to the standard sampling rate. Otherwise, the AP gradually reduces sampling rates to save energy.

## IX. Related Work

**Wi-Fi downclocking.** Several recent efforts have demonstrated the benefits and feasibility of downclocking Wi-Fi radios. E-Mili [3] first brings downclocking to design low-power Wi-Fi by reducing receiver's sampling rate during idle listening. E-Mili modifies packet preamble to ensure packet detection while downclocking. SloMo [4] leverages PHY sparsity in DSSS to decode 802.11b packets using downclocked radio. Enfold [5] extends SloMo to OFDM-based Wi-Fi. However, Enfold requires very high SNR (>25dB) and low-density modulation schemes to provide sufficient redundancy in constellation symbols. These approaches either require packet modifications [3], or large PHY redundancy that can hardly be guaranteed by today's Wi-Fi networks. Sampleless Wi-Fi overcomes this limitation by exploiting MAC redundancy, which can be easily satisfied by simple retransmission scheduling amendments.

**Sparse Recovery.** Our deocoding algorithm is also related to sparse FFT [10] and compressive sensing techniques [21], [22]. BigBand [10] utilizes sparse FFT [23] to allow multiple receivers of low sampling rates to perform GHz-wide realtime spectrum sensing. Similarly, compressive sensing techniques [21], [22] realize downclocking at receivers by adding an extra GHz analog mixer before a low rate ADC to perform high-speed complex analog matrix multiplications and analog mixing. Different from sparse FFT and compressive sensing, our decoding algorithm exploits the sparsity lying in the constellation map based on the properties of time shifting and the idea of rateless transmission.

**Power saving protocols.** 802.11 PSM [9] puts devices into a low-power sleep mode whenever they are in inactive periods. Many variants of PSM have been proposed to dynamically adjust sleep mode periods based on traffic patterns [24], [25], or extend the sleep mode to intervals between packets for devices with light workloads [26], [27]. However, the effectiveness of the PSM-based approaches is limited in real networks. First, the performance of these approaches heavily relies on traffic patterns, which are unpredictable in many cases [3], [4]. Additionally, the power used for contention and packet reception is not reduced. Finally, some popular frameworks, such as [28], [29], allows applications to override the PSM mechanism, forcing Wi-Fi chipset to stay awake to retain application services. Sampleless Wi-Fi can still reduce power consumption in these cases as it decreases the sampling rates across all Wi-Fi states.

Another class of protocols adds duty cycle mechanism into Wi-Fi to periodically turn off the radio. Cell2Notify [30] turns off Wi-Fi during idle states and leverages cellular radio to notify pending data for Wi-Fi. Shih et al. [31] implement a separated low-energy control channel to wake up radios for pending data while turning off radios by default for energy saving. Some other works duty cycling Wi-Fi radios based on Wi-Fi availability prediction. Rahmati et al. [32] estimate the probability distribution of Wi-Fi network conditions based on user's context information, and selectively turn on Wi-Fi or cellular interfaces to save energy. Similarly, Breadcrumbs [33] forecasts Wi-Fi connectivity based on user's historical trajecto-

ries. Instead of predicting Wi-Fi connectivity based on human habits, Blue-Fi [34] and FreeBee [35] leverage the low-energy Bluetooth interface to discover Wi-Fi networks.

Other proposals use power control to efficiently adjust power for transmission based on SNR conditions. Wi-Fi transceivers can perform data transmission using sub-mW energy under high SNR conditions. Unfortunately, studies [20], [36] have shown that energy consumption of power amplifier is quite limited compared to other components in Wi-Fi chipsets, including ADC/DAC. Wizizz [37] exploits the characteristics of channel bonding to propose an energy efficient bandwidth management scheme for IEEE 802.11ac. Wizizz realizes power saving by reducing unnecessary bandwidth, which is similar to power saving through sampling rate reduction in Sampleless Wi-Fi. The difference between Sampleless Wi-Fi and Wizizz lies in that Wizizz manages legacy bandwidths with a lower bound of a single channel bandwidth, while Sampleless Wi-Fi further reduces the sampling rates to sub-Nyquist rates to receive legacy packets.

**Rateless codes design.** Rateless codes can achieve optimal bitrate [13], [38]. In rateless codes designs, a transmitter continuously sends correlated packets until the receiver recovers the data by combining these packets. Recently, rateless codes have been extended to different applications, such as long distance transmission [39] and routing [40]. Sampleless Wi-Fi is inspired by rateless codes, while departing from them in that Sampleless Wi-Fi creates time shift induced constellations instead of modifying legacy packets.

## X. Conclusion

This paper presents Sampleless Wi-Fi, an architecture that provides reliable communications between legacy APs and low-power devices with various sampling rates. By exploiting the time shift effect on constellation mapping, Sampleless Wi-Fi leverages retransmission opportunities to decode legacy packets with sub-Nyquist rates under a wide range of channel quality conditions. The main merit of Sampleless Wi-Fi design is that it can be seamlessly integrated with existing Wi-Fi networks, and works reliably for various channel conditions and traffic types. Our experimental evaluation and trace-driven simulations validate Sampleless Wi-Fi in real Wi-Fi networks. We hope the design of Sampleless Wi-Fi can contribute the wireless community by improving the energy efficiency to mitigate the mismatch between high sampling rate required by Wi-Fi and the energy constraints of post-PC devices.

Sampleless Wi-Fi allows future low-end receivers to connect with existing Wi-Fi infrastructures to decode legacy packets at lower sampling rates under a wide range of SNRs. Sampleless Wi-Fi receivers can be seamlessly integrated into existing Wi-Fi networks with minimal modifications at APs. We believe that with these features, Sampleless Wi-Fi can provide some insights for future Wi-Fi receiver design.

## References


[1] *Android Wear Help: Connect to Wi-Fi*. [Online]. Available: https://support.google.com/androidwear/answer/6207505?hl=en

[2] W. R. Dieter, S. Datta, and W. K. Kai, "Power reduction by varying sampling rate," in *Proc. ACM ISLPED*, 2005.

[3] X. Zhang and K. G. Shin, "E-mili: energy-minimizing idle listening in wireless networks," in *Proc. ACM MobiCom*, 2011.

[4] F. Lu, G. M. Voelker, and A. C. Snoeren, "Slomo: Downclocking wifi communication." in *USENIX NSDI*, 2013.

[5] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren, "Enfold: downclocking ofdm in wifi," in *Proc. ACM MobiCom*, 2014.

[6] M. Luby, "Lt codes," in *FOCS*, 2002.

[7] "Part 11: Wireless lan medium access control (mac) and physical layer (phy) amend-ment 8: Medium access control (mac) quality of ser-vice enhancements," *IEEE Std. 802.11e WG*, Nov. 2005.

[8] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benkö, J. Chiang, A. C. Snoeren, S. Savage, and G. M. Voelker, "Automating cross-layer diagnosis of enterprise wireless networks," in *Proc. ACM SIGCOMM*, 2007.

[9] "Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std. 802.11*, 2007.

[10] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "Bigband: Ghz-wide sensing and decoding on commodity radios," in *Proc. IEEE INFOCOM*, 2014.

[11] R. G. Lyons, *Understanding digital signal processing*. Pearson Education, 2010.

[12] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, 1973.

[13] A. Gudipati and S. Katti, "Strider: Automatic rate adaptation and collision handling," in *Proc. ACM SIGCOMM*, 2011.

[14] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, "Frequency-aware rate adaptation and mac protocols," in *Proc. ACM MobiCom*, 2009.

[15] R. Mittal, A. Kansal, and R. Chandra, "Empowering developers to estimate app energy consumption," in *Proc. ACM MobiCom*, 2012.

[16] A. Schulman, D. Levin, and N. Spring, "On the fidelity of 802.11 packet traces," in *PAM*, 2008.

[17] L. E. Li, K. Tan, H. Viswanathan, Y. Xu, and Y. R. Yang, "Retransmission≠ repeat: simple retransmission permutation can resolve overlapping channel collisions," in *Proc. ACM MobiCom*, 2010, pp. 281–292.

[18] A. Kamerman and L. Monteban, "Wavelan®-ii: a high-performance wireless lan for the unlicensed band," *Bell Labs Techn. J.*, vol. 2, no. 3, pp. 118–133, 1997.

[19] *Minstrel Rate Adaptation Algorithm Documentation*. [Online]. Available: http://madwifiproject.org/

[20] D. Halperin, B. Greenstein, A. Sheth, and D. Wetherall, "Demystifying 802.11 n power consumption," in *Proc. USENIX HotPower*, 2010, p. 1.

[21] J. Laska, W. Bradley, T. W. Rondeau, K. E. Nolan, and B. Vigoda, "Compressive sensing for dynamic spectrum access networks: Techniques and tradeoffs," in *Proc. IEEE DySPAN*, 2011.

[22] Y.-C. Chen, L. Qiu, Y. Zhang, Z. Hu, and G. Xue, "Robust network compressive sensing," in *Proc. ACM MobiCom*, 2014.

[23] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Nearly optimal sparse fourier transform," in *Proc. ACM STOC*, 2012.

[24] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," in *Proc. ACM MobiCom*, 2002.

[25] M. Anand, E. B. Nightingale, and J. Flinn, "Self-tuning wireless network power management," *Springer Wireless Netw.*, vol. 11, no. 4, pp. 451–469, 2005.

[26] J. Liu and L. Zhong, "Micro power management of active 802.11 interfaces," in *Proc. ACM MobiSys*, 2008.

[27] F. R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices," in *Proc. ACM MobiSys*, 2010.

[28] *Android Code Project: Add WIFI-MODEFULL-HIGH-PERF (or similar) to official SDK*. [Online]. Available: http://code.google.com/p/android/ issues/detail?id=15549

[29] *Android Developer Reference: WifiManager*. [Online]. Available: http://developer.android.com/reference/android/net/wifi/WifiManager.html.

[30] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless wakeups revisited: energy management for voip over wi-fi smartphones," in *Proc. ACM MobiSys*, 2007, pp. 179–191.

[31] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: an event driven energy saving strategy for battery operated devices," in *Proc. ACM MobiCom*, 2002, pp. 160–171.

[32] A. Rahmati and L. Zhong, "Context-for-wireless: context-sensitive energy-efficient wireless data transfer," in *Proc. ACM MobiSys*, 2007, pp. 165–178.

[33] A. J. Nicholson and B. D. Noble, "Breadcrumbs: forecasting mobile connectivity," in *Proc. ACM MobiCom*, 2008, pp. 46–57.

[34] G. Ananthanarayanan and I. Stoica, "Blue-fi: enhancing wi-fi performance using bluetooth signals," in *Proc. ACM MobiSys*, 2009, pp. 249–262.

[35] S. M. Kim and T. He, "Freebee: Cross-technology communication via free side-channel," in *Proc. ACM MobiCom*, 2015, pp. 317–330.
[36] F. I. Di Piazza, S. Mangione, and I. Tinnirello, "On the effects of transmit power control on the energy consumption of wifi network cards," in *Springer QSHINE,*, 2009, pp. 463–475.
[37] O. Lee, J. Kim, and S. Choi, "Wizizz: Energy efficient bandwidth management in ieee 802.11 ac wireless networks," in *Proc. IEEE SECON*, 2015, pp. 136–144.
[38] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah, "Spinal codes," in *Proc. ACM SIGCOMM*, 2012.
[39] W. Du, Z. Li, J. C. Liando, and M. Li, "From rateless to distanceless: Enabling sparse sensor network deployment in large areas," in *Proc. ACM SenSys*, 2014.
[40] Z. Li, W. Du, Y. Zheng, M. Li, and D. Wu, "From rateless to hopless," in *Proc. ACM MobiHoc*, 2015.