

# Wi-Fi Teeter-Totter: Overclocking OFDM for Internet of Things

Wei Wang<sup>†</sup>, Shiyue He<sup>†</sup>, Lin Yang<sup>‡</sup>, Qian Zhang<sup>‡</sup>, Tao Jiang<sup>†</sup>

<sup>†</sup> School of Electronic Information and Communications, Huazhong University of Science and Technology

<sup>‡</sup> Department of Computer Science and Engineering

Hong Kong University of Science and Technology

Email: {weiwangw, shiyue\_he, taojiang}@hust.edu.cn, {lyangab, qianzh}@hust.edu.cn

**Abstract**—The conventional high-speed Wi-Fi has recently become a contender for low-power Internet-of-Things (IoT) communications. OFDM continues its adoption in the new IoT Wi-Fi standard due to its spectrum efficiency that can support the demand of massive IoT connectivity. While the IoT Wi-Fi standard offers many new features to improve power and spectrum efficiency, the basic physical layer (PHY) structure of transceiver design still conforms to its conventional design rationale where access points (AP) and clients employ the same OFDM PHY. In this paper, we argue that current Wi-Fi PHY design does not take full advantage of the inherent asymmetry between AP and IoT. To fill the gap, we propose an asymmetric design where IoT devices transmit uplink packets using the lowest power while pushing all the decoding burdens to the AP side. Such a design utilizes the sufficient power and computational resources at AP to trade for the transmission (TX) power of IoT devices. The core technique enabling this asymmetric design is that the AP takes full power of its high clock rate to boost the decoding ability. We provide an implementation of our design and show that it can reduce the IoT’s TX power by boosting the decoding capability at the receivers.

## I. INTRODUCTION

We are entering the post-PC era where an ever-larger variety of smart Internet-of-things (IoT) devices including wearables and smart sensors are increasing the demands for low power communication technologies. Wi-Fi has recently become a contender for this regime due to its compatibility with IP networks and wide deployments of hotspots. Growing numbers of wearables in the market have enabled Wi-Fi connection to the Internet without relying on gateways. Many commercial smart watches including Apple Watch have already equipped with built-in Wi-Fi chipsets. Google’s latest Android Wear 2.0 release [1] allows Android smartwatches to directly connect to Wi-Fi networks.

With the proliferation of sensor-equipped smart objects in streets, homes, and offices, IoT connectivity is envisioned to be massive [2], [3]. Thus, Wi-Fi for IoT devices should be not only low power but also spectrum efficient. As a mature and spectrum efficient multiplexing access technology widely adopted in the latest Wi-Fi standards, OFDM continues its adoption in the new Wi-Fi standard for IoT, i.e., IEEE 802.11ah [4], which reuses OFDM frame format that conforms to IEEE 802.11ac to ensure spectrum efficiency.

A number of recent research efforts have been devoted to reducing the power consumption of OFDM-based Wi-Fi

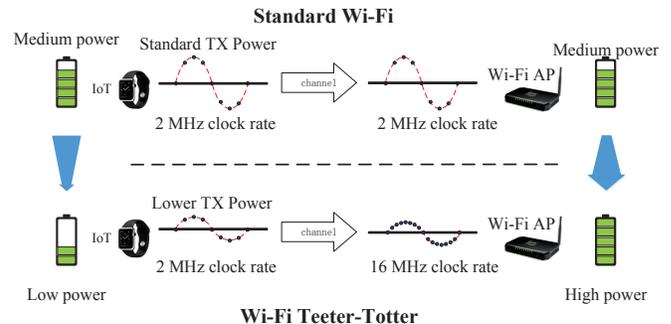


Fig. 1. The concept of Wi-Fi Teeter-Totter. A IoT device lowers its TX power and pushes all decoding and energy burdens to the AP side. As the compensation for decreased TX power, the AP takes full advantage of its high clock rate.

communications. Downclocking receivers’ radios is proposed to reduce the power consumption while receiving packets [5], [6] or idle listening [7], [8]. Efficient sleep modes [9], [10] are proposed to reduce the power consumption of conventional IEEE 802.11 nodes by allowing them to enter extremely low power state during idle listening. Despite these growing attempts and extensive efforts, most of them have focused on energy-efficient data reception. A variety of IoT applications require data-rich sensors such as cameras and microphones to frequently upload sampled data to servers, which incurs a large amount of energy consumption for uplink transmission.

In this paper, we argue that the fundamental hurdle for energy efficient uplink transmission lies in the transceiver design. Existing Wi-Fi transceivers are originally designed for symmetric nodes with equal hardware capabilities and power constraints. This assumption no longer stands for IoT applications, where a Wi-Fi AP is much more powerful and almost energy-unconstrained compared to IoT devices. This evokes a similar picture in cellular networks, where base stations are equipped with  $100\times$  sensitive RF front-end and have 20 dB more transmission (TX) power than mobile clients. Analogous to the hardware asymmetry in cellular networks, APs are equipped with Wi-Fi chipsets that can support up to 160 MHz bandwidth in IEEE 11ac/11ax, while IoT devices normally support only 1-2MHz bandwidth [4]. Thus, there is

significant potential to exploit asymmetric PHY configurations to enable low power uplink transmission without undermining decoding performance.

Our line of attack starts from the rationale that we trade the computing and energy resources of APs for the TX power of IoT devices in a *teeter-totter* fashion, that is, we allow IoT devices to transmit uplink packets using the lowest power while pushing all the decoding burdens to the AP side, as illustrated in Figure 1. In particular, the AP takes full advantage of its clock rate that are tens of times higher than those of IoT's to decode packets with low signal-to-noise ratio (SNR). IoT devices reap benefits from such an asymmetric design by configuring TX power lower than the minimum power required for decoding. On the other hand, the AP with enough computing capabilities and constant power source can afford the extra cost induced by overclocking. Such an asymmetric design trades resources that are cheap to AP for power consumptions that are expensive to IoTs. Such a design completely conforms to Wi-Fi protocols, and thus can be readily integrated with existing standards. The only changes are standalone update of computational logics and RF settings at APs.

A key challenge in realizing the teeter-totter design is how to effectively leverage overclocking in OFDM to decode conventionally undecodable packets under low SNR conditions. Our fundamental insight is that APs yield correlated signals by exploiting the *time shift* effect between overclocked samples. Specifically, when the AP sets its analog-to-digital converter (ADC) clock at a much higher rate than IoT devices, it yields multiple interpolated samples from one transmitted sample. These interpolated samples can be considered as time-shifted versions of the transmitted sample. When transformed into the frequency domain, time-shifted samples result in different phase rotations at different subcarriers. Thus, we can leverage this phase rotation effect to combine these samples for packet decoding. Another challenge stems from the lack of knowledge in modeling the noise distributions between these redundant samples. Instead of blindly reusing conventional decoders, we turn to a data-driven approach. In particular, we build a noise map from preambles without making distribution assumptions. Then, we combine redundant samples by employing a maximum likelihood (ML) decoder based on the joint probability of these samples.

We implement the above teeter-totter design, referred to as T-Fi, on the GNURadio/USRP platform. Evaluation results validate T-Fi in reliably receiving and decoding Wi-Fi packets at low SNR across a wide range of scenarios. Furthermore, T-Fi can reduce the IoT's TX power when the AP is overclocked by a factor of up to 8, which is still lower than the rate used for IEEE 802.11a/g/n.

The contributions of this paper are summarized below.

- We provide a thoughtful study towards enabling low-power OFDM transmissions for IoT Wi-Fi standards. Our solution can be seamlessly integrated into existing Wi-Fi standards without modifying the legacy frames or protocols.

- We explore the fundamental structure of overclocked reception in OFDM, and propose a reception pipeline to decode legacy packets at lower SNRs than the conventional transceivers. The key underlying technique is a new decoding algorithm that exploits the time shift effects in oversampled signals.
- We build a full prototype of T-Fi and quantify the merits of our design in a wide range of scenarios.

The remainder of the paper is structured as follows. We begin in Section II with the design motivation. Section III analyzes the effect of overclocking in OFDM, which is the underpinnings of our design. Section IV elaborates the detailed reception pipeline of our design. System implementation and experimental evaluation are introduced in Section V and Section VI, respectively. Section VII gives a brief survey of related work, followed by conclusion in Section VIII.

## II. MOTIVATION OF TRANSCEIVER ASYMMETRY

In this section, we ask and answer questions to explore practical and suitable transceiver architectures. Through this exploration, we hope to convince readers that the teeter-totter design is a practical design in that it elegantly fits the architecture of IoT Wi-Fi while imposing affordable costs.

### A. IoT Scenarios and Requirements

It is envisioned that the number of smart devices that need to wirelessly share data or access the Internet is growing exponentially. To embrace the coming wave of *massive numbers* of wearables, driverless cars, smart sensors, IEEE 802.11ah was announced in 2016 to tailor Wi-Fi protocols specially for these IoT devices. The target envisioned in this IoT Wi-Fi standard is to challenge Bluetooth and cellular networks by achieving the best of both worlds: to enable *long-distance* communications using relatively *low amounts of power*. Specifically, IEEE 802.11ah defines the following requirements: (i) more than a fourfold number of devices supported by one AP compared to legacy Wi-Fi, (ii) doubled transmission range compared to legacy Wi-Fi links, and (iii) low energy consumption with guaranteed data rates of at least 100 Kb/s.

To meet the above requirements, IEEE 802.11ah introduces a bundle of new features in PHY/MAC. Particularly, the new standard reuses the formerly-adopted OFDM waveform consisting of 32/64 subcarriers while tailoring it to a narrowband channel of 1 MHz or 2 MHz in 900 MHz unlicensed spectrum. OFDM is used to maintain spectrum efficiency for massive connectivity, and narrowband transmission at lower frequencies are used to facilitate low-power, long-range transmissions. In addition, a set of MAC mechanisms is designed to reduce the power consumption in channel access.

### B. Hardware Asymmetry

Although the new standard specifies new PHY and MAC techniques and features to meet the requirements of IoTs, Wi-Fi transceivers still work in a conventional fashion: today's Wi-Fi radios work in a symmetric manner in that the transmitter

and receiver are configured with the same clock rate. This symmetric transceiver design has worked well over the past decade as both sides are expected to be (more or less) equally powerful in conventional scenarios. While this assumption no longer stands in IoT scenarios where the disparity between the hardware capabilities of APs and IoTs makes such a design very inefficient. It is either an overwhelming burden for IoTs to match the radio configurations of APs, or makes APs largely under-utilize their hardware capabilities to shoehorn the proper settings of IoTs' radios.

The above situation evokes a similar picture in cellular networks, where base stations are much more powerful than mobile stations and are designed to support thousands of hardware-constrained mobile devices. Cellular networks equip base stations with higher TX power, RF sensitivity, and large form-factor antennas to compensate the hardware gap. The wisdom of cellular networks motivates us to design an asymmetric PHY that fully reap the benefits of the hardware gap between APs and IoTs.

Our target is to explore an asymmetric transceiver design that can better fit the IoT scenarios and be seamlessly integrated into existing and future OFDM-based Wi-Fi protocols. The fundamental insight is that APs have already equipped with high clock-rate radios to support high-speed Wi-Fi such as IEEE 802.11ac/ax, and the clock rates are tens of times to those of IoTs'. When APs use such overclocked rates to sample IoT's transmissions, a large amount of redundancy between these samples can be used to boost the performance of packet reception, which in turn relaxes TX power requirements imposed on IoTs.

### III. EXPLOITING OVERCLOCKING OPPORTUNITIES

At the core of the asymmetric transceiver design, we exploit the redundancy in OFDM signals sampled by overclocked radios. To this end, we discuss the opportunities in retaining the OFDM samples at an overclocked rate and utilizing them to improve symbol decoding. We start by analyzing the correlations among these samples.

In a standard OFDM system, as illustrated in Figure 2, one OFDM symbol contains a sequence of bits that are modulated into a set of lattice points  $X[f]$  on orthogonal subcarriers in the frequency domain. The information-bearing  $X[f]$ . For OFDM transmissions, inverse fast Fourier transform (IFFT) are used to transform  $X[f]$  into a time-domain sequence

$$x[n] = \frac{1}{F} \sum_{f=0}^{F-1} X[f] e^{j2\pi fn/F}, n = 0, \dots, F-1,$$

where  $F$  is the number of subcarriers. Note that we omit cyclic prefix for simple illustration. The discrete-time signal  $x[n]$  passes an digital-to-analog converter (DAC) and a pulse shaping filter to transform itself into a continuous-time signal in baseband form, which is written as

$$x(t) = \sum_{f=0}^{F-1} x[n] p(t - fT_s),$$

where  $p(t)$  is the impulse response of the pulse-shaping filter and  $1/T_s$  is the baud rate.  $x(t)$  is transmitted and propagates a wireless channel with impulse response  $h(t)$ . A receiver receives the signals with a matched filter  $p(-t)$ , and obtain

$$y(t) = \sum_{n=0}^{F-1} x[n] p(t - fT_s) * h(t - fT_s) * p(fT_s - t) + w(t),$$

where  $*$  denotes convolution, and  $h(t)$ ,  $w(t)$  are the impulse response of the channel and complex Gaussian noise, respectively. In the presence of a time-dispersive channel and additive noise, the received continuous time-domain baseband signal  $y(t)$  can be expressed as

$$y(t) = \frac{1}{F} \sum_{f=0}^{F-1} X[f] H(f) e^{j\frac{2\pi f t}{T}} + w(t),$$

where  $H(f)$  is the channel frequency response at subcarrier  $f$ , and  $T$  is the symbol duration.

Now we consider a concrete example in which  $F = 64$  subcarriers are used to convey information. When using the same clock rate to sample received signal  $y(t)$ , the sampling instances are at  $t = \frac{n}{64}T$ . Thus, the receiver yields

$$y_1[n] = \frac{1}{64} \sum_{f=0}^{63} X[f] H(f) e^{j\frac{2\pi f n}{64}} + w_1(n).$$

When the receiver doubles the clock rate, the number of samples in each FFT segment is 128. As such, the sampling instances are at  $t = \frac{n}{128}T$ , and the time domain sample sequence is expressed as

$$\begin{aligned} y[n] &= \frac{1}{64} \sum_{f=0}^{63} X[f] H(f) e^{j\frac{2\pi f n}{128}} + w(n), n = 0, \dots, 127 \\ &= \frac{1}{64} \sum_{f=0}^{63} \left( X[f] H(f) e^{j\frac{2\pi f}{128} 2n} + X[f] H(f) e^{j\frac{2\pi f}{128} (2n+1)} \right) \\ &\quad + w_1(2n) + w_1(2n+1), n = 0, \dots, 63 \\ &= y_1[n] + y_2[n] + w_2(n), n = 0, \dots, 63, \end{aligned} \tag{1}$$

where  $y_2[n] = \frac{1}{64} \sum_{f=0}^{63} X[f] H(f) e^{j\frac{2\pi f}{64} (n+1/2)}$  and  $w_2(n)$  is the new noise sequence in oversampled signals. We see that the oversampled signal  $y[n]$  consists of two polyphase components  $y_1[n]$  and  $y_2[n]$ . Notice that  $y_2[n]$  is a time-shifted version (delayed by a half sample) of  $y_1[n]$ . The frequency response of  $y_1[n]$  can be derived by performing FFT.

$$\begin{aligned} Y_1[l] &= \sum_{n=0}^{63} y_1[n] e^{-j\frac{2\pi n l}{64}}, l = 0, \dots, 63 \\ &= \frac{1}{64} \sum_{n=0}^{63} \sum_{f=0}^{63} X[f] H(f) e^{j\frac{2\pi f n}{64}} e^{-j\frac{2\pi n l}{64}} + W_1(l) \\ &= X[l] H(l) + W_1(l), \end{aligned}$$

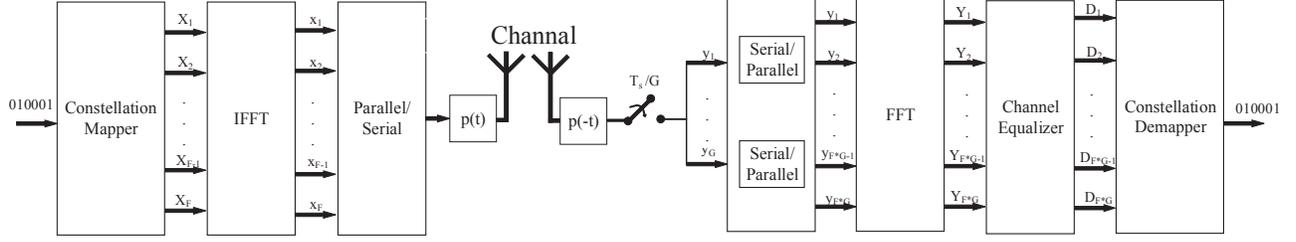


Fig. 2. OFDM schematic under overclocking.

where  $W_1(l)$  is the noise expressed in the frequency domain. Similarly, the frequency response of  $y_2[n]$  is expressed as

$$\begin{aligned} Y_2[l] &= \sum_{n=0}^{63} y_2[n] e^{-\frac{j2\pi nl}{64}}, l = 0, \dots, 63 \\ &= \frac{1}{64} \sum_{n=0}^{63} \sum_{f=0}^{63} X[f] H(f) e^{j\frac{2\pi f}{64}(n+1/2)} e^{-\frac{j2\pi nl}{64}} + W_2(l) \\ &= X[l] H(l) e^{-\frac{j\pi l}{64}} + W_2(l). \end{aligned}$$

Ignoring the noises,  $Y_2[l]$  is a phase-shifted version of  $Y_1[l]$ , and the amount of phase rotation is linear to the subcarrier frequency  $l$ .

Hence, doubling the receiver's clock rate yields two receptions, which transforms the system into a single-input-multiple-output (SIMO) system. When we employ  $G$ -fold clock rate at the receiver, we can obtain  $G$  phase-shifted versions of  $Y_1[l]$  described as

$$Y_g[l] = X[l] H(l) e^{-\frac{j2\pi gl}{64G}} + W_g(l), g = 0, \dots, G-1. \quad (2)$$

Figure 3 shows the phase shifts across all subcarriers in a real Wi-Fi packet when received at eight-fold clock rate using USRP testbeds. The predictable phase shifts in  $Y_g[l]$  can be easily compensated to obtain  $G$  copies of the transmitted signal. We can combine these copies to improve the received signal strength while amortizing noise, and thereby enhance the decoding capability under low SNR conditions.

#### IV. ASYMMETRIC TRANSCEIVER DESIGN

In this section, we describe our design of T-Fi, an asymmetric transceiver architecture for IoT Wi-Fi. T-Fi can fully interoperate with standard IEEE 802.11 devices, with no modifications to existing protocols. T-Fi leverages the high clock rate of APs to enable low power transmissions for IoT devices. This section elaborates the detailed reception pipeline design that embraces this design rationale.

##### A. Frame structure and reception pipeline

A Wi-Fi frame starts with a preamble prepending payload to perform time synchronization, carrier frequency offset (CFO) compensation, and channel estimation. In particular, a preamble consists of a Short Training field (STF) and a

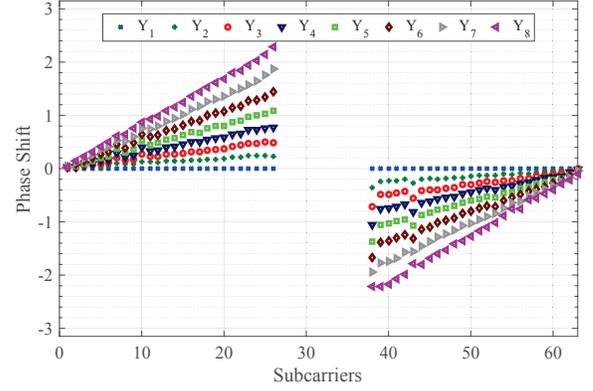


Fig. 3. Phase shift induced by oversampling a real Wi-Fi packet. Phases are normalized according to the phases of  $Y_1$ . Subcarriers 0-25, 38-63 carrying payload data are compared.

Long Training Field (LTF). The legacy STF contains two OFDM symbols that are comprised of ten repetitions of a 16-sample sequence, and the LTF contains two identical 64-sample (80-sample including cyclic prefix) OFDM symbols. A PHY header, known as signaling field (SIG), containing modulation and coding scheme (MCS) information, duration, and service field, sits between the preamble and payload. The preamble and the first 24 bits of the PHY header are encoded using the lowest MCS mode to ensure their correct reception, while the rest can be encoded using different MCS modes based on a certain rate adaptation algorithm.

Under overclocking settings, a receiver increases the sampling rate of the ADC by  $G$  folds. The incoming analog signals are oversampled and converted into  $G$  baseband streams, which are fed into the synchronization block to identify the start of a Wi-Fi frame. Meanwhile, the CFO between the sender and the receiver is estimated and compensated using preamble sequences to eliminate the effect of the disparity between clock oscillator frequencies. Since there may still be residual frequency offset, phase compensation is employed to correct the phase rotation throughout the entire payload. In Wi-Fi, four subcarriers are used as pilots to estimate phase rotation in each data symbol. Then, the receiver performs FFT

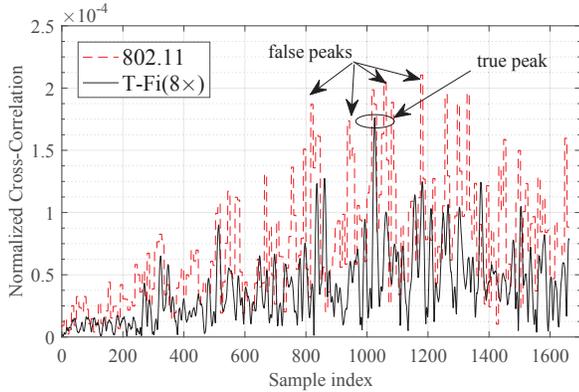


Fig. 4. Cross correlation responses at various clock rates. We set up a USRP link to capture raw data samples of a real Wi-Fi packet at  $8\times$  and  $1\times$  Nyquist rates, respectively.

on the samples to obtain their frequency domain values. After performing FFT, the receiver estimates channel responses by comparing the received values in LTF and the actual values transmitted. Finally, the receiver decodes the bits in the payload symbols.

In what follows, we systematically exploit the merits as well as issues brought by overclocking, and propose corresponding techniques to optimize the packet reception performance.

### B. Timing Synchronization

Timing synchronization aims to detect the presence of a Wi-Fi packet and then identify the start of Wi-Fi symbols. Wi-Fi receivers take advantage of the periodicity in STF to detect the Wi-Fi preamble. It measures the energy of the sampled signal  $\mathbf{y}$ :  $P = \sum_{k=1}^L |y[k]|^2$ , where  $L$  is the measuring window. If  $P_s$  is higher than a given threshold, it then performs auto correlation to check whether the incoming signal is a Wi-Fi frame. Specifically, it tracks the auto correlation of the signal with  $d$  delayed samples:

$$c_{\text{auto}} = \sum_{k=1}^L y[n+k]r^*[n+k-d].$$

$d$  is normally set to be 16, which is the period of one short sequence in STF. For a Wi-Fi packet,  $c_{\text{auto}}$  spikes and forms a plateau. Our experiments show that even at very low SNR, the plateau is still clear to be separated from noise. Therefore, we configure normal clock rate for packet detection to avoid unnecessary energy consumption in idle listening and carrier sensing.

After successfully detecting a Wi-Fi packet, the receiver precisely locates the boundary between OFDM symbols. While the IEEE 802.11 specifications do not mandate any specific algorithm, a typical synchronization algorithm is to use the cross-correlation property of the LTF. In particular, the receiver correlates the received signal with the transmitted time-domain LTF sequence. However, when the SNR is poor, cross correlation cannot cancel out noise, thereby resulting in multiple false peaks. We exploit correlations in oversampled

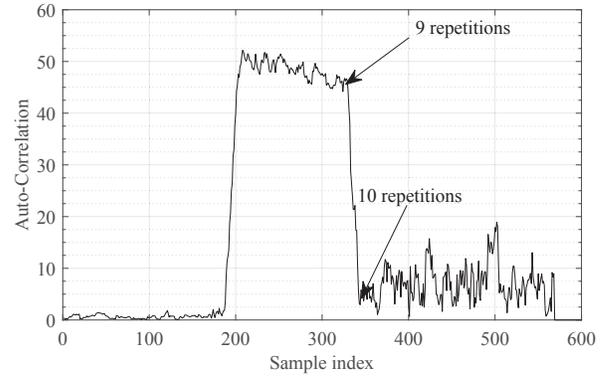


Fig. 5. Auto correlation response.

LTF signals to overcome this predicament. The standard LTF consists of 52 subcarriers with the identical magnitude and 0 or  $\pi$  phase. When the LTF is sampled by an overclocked ADC, the correlation between oversampled data can be described by Eq. 2. Hence, the receiver can treat all the samples as a new LTF, which has stronger correlation properties. Figure 4 illustrates the merits of using the oversampled LTF for synchronization. Under low SNR conditions, the cross correlation result of a standard LTF produces multiple comparable peaks, which makes the receiver easily aligns to a wrong peak. The cross correlation result of an oversampled LTF produces a single highest peak corresponding to the full alignment of OFDM symbols.

We set normal clock rate for idle listening to avoid unnecessary energy consumption. After packet detection, the receiver switches to the overclocking mode, which incurs latency for clock switch. As the receiver keeps the same frequency synthesizer and the center frequency of its analog circuit, the latency comes from the digital phase-locked loop (PLL) stabilization. Wi-Fi radios take merely several microseconds (e.g.,  $8 \mu\text{s}$  in MAXIM2831 [11]) to stabilize PLL. As the state-of-the-art IoT Wi-Fi, i.e., IEEE 802.11ah, shrinks the conventional 20 MHz bandwidth to 2 MHz while retaining the same number of subcarriers, the duration of one OFDM symbol is extended to  $40 \mu\text{s}$ . Both STF and LTF last  $80 \mu\text{s}$ , which leaves enough time for clock switch. As illustrated in Figure 5, which is tested with SNR=10 dB, we observe that the first nine repetitions are enough to produce a plateau for packet detection. Thus, we take the first nine repetitions in STF to perform auto correlation while leaving the last repetition of STF for clock switch. In our experiment results, we show that it does not need to extra packet loss and the whole timing synchronization perform is boosted compared to the standard algorithm under low SNR conditions.

### C. Frequency Offset Compensation

CFO varies over time and must be estimated and compensated for each frame. In practice, the received baseband signal, instead of being centered at DC (0 Hz), is centered at

a frequency offset  $\Delta f$ .

$$y_{\text{CFO}}(t) = y(t)e^{j\frac{2\pi\Delta f t}{F_s}},$$

where  $F_s$  is the sampling frequency. CFO induces phase rotation over time that not only undermines the payload decoding but also affects the phase correlation among  $Y_g[l]$ . As our overlocking design relies on the phase correlation, CFO must be precisely estimated and calibrated.

Standard Wi-Fi receiver estimates CFO by comparing the phase rotation between the two identical OFDM symbols in LTF. Such an estimation is accurate enough for normal packet decoding, while the residual offset cannot be neglected when using the phase correlation among oversampled data. We make a fine-grained calibration in the following data symbols by exploiting the CFO effect among oversampled sequences  $y_g[n]$ . Let  $\mathbf{y}_g = [y_g[0], \dots, y_g[N-1]]^T$  denotes  $g$ th copy of the oversample signals, and the received data values in frequency domain  $\mathbf{d} = [X[0]H(0), \dots, X[N-1]H(N-1)]^T$ . FFT is expressed in the form of a  $N \times N$  matrix  $\mathbf{F}$ , where each entry  $f_{ij} = e^{j(i-1)(j-1)}$ . The effect of CFO on the first copy can be expressed as  $\mathbf{P} = \text{diag}\left(1 e^{j\frac{\Delta f}{F_s}} \dots e^{j\frac{\Delta f(N-1)}{F_s}}\right)$ . The phase shifts due to overlocking can be described by  $\mathbf{O}_g = \text{diag}\left(1 e^{-j\frac{2\pi g}{NG}} \dots e^{-j\frac{2\pi g(N-1)}{NG}}\right)$ . Then, the  $g$ th copy of the oversample signals  $\mathbf{y}_g$  can be expressed as

$$\mathbf{y}_g = e^{j\frac{2\pi\Delta f g}{F_s G}} \mathbf{P} \mathbf{F} \mathbf{O}_g \mathbf{d} + \mathbf{W}_g. \quad (3)$$

In the absence of noise, we obtain the following the relationship

$$\mathbf{F}^H \mathbf{P}^H \mathbf{y}_g = e^{-j\frac{2\pi\Delta f g}{F_s G}} \mathbf{O}_g^H \mathbf{F}^H \mathbf{P}^H \mathbf{y}_0, \quad (4)$$

where  $(\cdot)^H$  is the conjugate transpose of a matrix. It is intuitive to find the unique  $\Delta f$  that satisfying the above equation. In the presence of noises, we minimize the sum of distances between the left and the right hands of Eq. 4 for all  $g > 0$ . We employ ML estimator to derive  $\Delta f$ .

#### D. Decoding

After timing synchronization and frequency offset compensation, channel response needs to be understood before packet decoding. Conventional Wi-Fi receivers utilize the two training symbols in LTF to estimate the channel response at each subcarrier. As the data bits encoded on each subcarrier in LTF are pre-known by the receiver, the channel response is obtained by comparing the received data values and the transmitted ones. Based on the estimated channel response, the receiver removes the effect of wireless channel on the following OFDM symbols to extract the transmitted data values. Noises are assumed to be independent and Gaussian distributed, and thus the receiver demodulates each data value by mapping it to the nearest lattice point in the constellation map. We extend the above rationality to the overlocking case by averaging multiple copies of compensated samples before feeding them into the standard ML decoder.

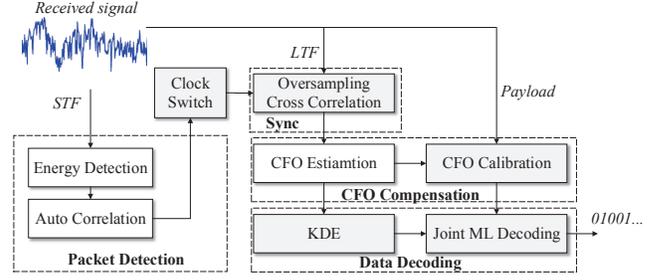


Fig. 6. System architecture of T-Fi.

#### E. Putting Everything Together

An overview of T-Fi system architecture is shown in Figure 6. T-Fi leverages the AP's high clock rate and processing capability to grant lower TX power for IoT devices. The reception pipeline conforms to the logic of a standard Wi-Fi receiver but is tailored to the asymmetric design to make the most of the redundancy induced by overlocking. An AP uses the first nine repetitions of the STF for packet detection and then switches to the high clock rate mode within the STF reception within the last one repetition. As the symbol length in IoT Wi-Fi packets is  $10\times$  of that in conventional Wi-Fi packets, it leaves enough time for the state-of-the-art chips to switch and stabilize the clock. Under the overlocking setting, the AP yields redundant LTF and payload samples, and exploits the correlations among these samples to boost the performance of synchronization and CFO estimation under low SNR conditions. Then synchronization algorithm is performed on the LTF, while CFO estimation is performed on the LTF and then calibrated in the following data symbols in the payload. Multiple copies of oversampled samples are averaged and fed to the standard ML decoder.

#### V. IMPLEMENTATION

T-Fi can be realized in the existing OFDM PHY with no hardware changes. We prototype T-Fi on top of the OFDM implementation on the GNURadio/USRP platform. We implement the entire PHY design specified in Section IV directly in the USRP Hardware Drive (UHD). We use USRP B210 nodes connecting to PCs with Intel i7 quad-core processor and 8 GB memory for the testbed setup. Nodes in our experiments are configured to operate in the 2.4-2.5 GHz or 900 MHz range.

The Transmitter is configured to continuously send Wi-Fi packets conforming to IEEE 802.11ah PHY format. It operates on a 2 MHz or 1 MHz channel, of which 52 subcarriers are configured to carry data values while 4 subcarriers are pilot tones. The symbol duration is  $40 \mu s$ . We adopt the legacy PHY layer convergence procedure (PLCP) format of IEEE 802.11ah, where the PLCP preamble consists of 2 STF OFDM symbols and 2 LTF OFDM symbols. There are 10 MCSs starting from 1/2 BPSK to 5/6 256QAM. Due to hardware

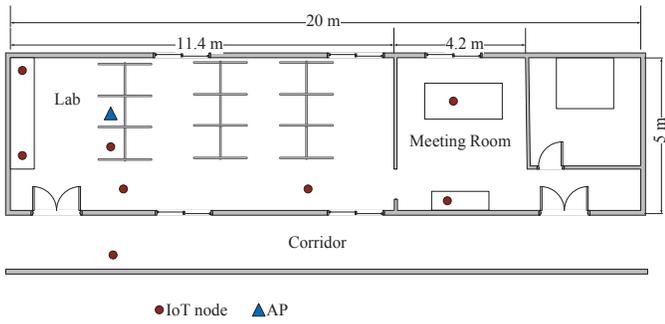


Fig. 7. Experimental floorplan.

limitations, we only implement BPSK, QPSK, 16QAM, and 64QAM modulations.

## VI. EVALUATION

In this section, we present a detailed experimental evaluation of T-Fi. Our experiments center around two questions: (i) How much decoding performance improvement can T-Fi provide in real wireless environments under proper overclocking settings? (ii) How much TX power can T-Fi save without compromising the decoding performance? To answer these questions, we conduct a set of experiments to evaluate the performance of synchronization, decoding, the overall packet reception, as well as TX power.

### A. Experimental Setup

We conduct our experiments using USRP B210 nodes deployed in an indoor environment with metal/wood shelves and stone walls. Figure 7 illustrates the floorplan of the environment, where nodes are placed in a lab, a meeting room, a hallway, and a corridor. We control the TX power by adjusting the programmable-gain amplifier (PGA) in USRP. As such, we sweep the SNR range from 9 dB to 35 dB. For each setting, a USRP node sends approximate 3000 packets. For each setup, we vary the clock rates of the receiver from the normal clock rate ( $1\times$ ) to overclocked rates ( $2\times$ ,  $4\times$ ,  $8\times$ ). Note that the clock rate of IEEE 802.11ac/ax receiver is  $10\text{--}80\times$ . We use the standard IEEE 802.11 packet reception at the normal clock rate as the baseline for comparison.

### B. Synchronization

First, we show that T-Fi addresses the synchronization issue under low SNR conditions. Recall that under low SNR conditions, conventional cross correlation algorithm cannot cancel out noise and yields multiple false peaks. We instead exploit the inherent correlation in the oversampled LFT signals to enhance the correlation property. Since we cannot directly obtain the ground truth of synchronization, we capture raw USRP samples and conduct offline analysis. In particular, we log raw samples received by a USRP node and feed them into an emulator to rehearse the synchronization process. We run the synchronization T-Fi at multiple clock rates ( $2\times$ ,  $4\times$ , and

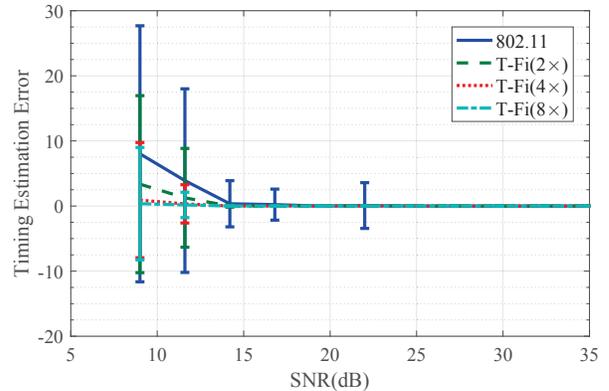


Fig. 8. Synchronization error.

$8\times$  clocking rates) and the standard synchronization algorithm at the normal clock rate ( $1\times$  clock rate).

Figure 8 compares the timing estimation error, which is measured in number of samples. Error bars show the standard deviation of time synchronization offset. Under all SNR conditions, synchronization error diminishes with the increment of clock rates. The standard synchronization algorithm works well for  $\text{SNR} > 15$ , while the synchronization error grows substantially when SNR is low (9 dB - 12 dB). Empowered by overclocking, T-Fi largely reduces the synchronization error. When  $\text{SNR}=9$  dB, the average synchronization error at  $8\times$  clock rate is 0.91, which is merely 13% of the average error at the standard clock rate. These results confirm the merit of using over-clocking to do time synchronization.

### C. Decoding

#### Decoding performance under various SNR conditions.

Our second set of experiments evaluates the decoding performance at different MCS modes under various environments and SNR conditions. To focus on the decoding performance, we use the same synchronization algorithm at the same clock rate for all decoders. In particular, we set  $8\times$  clock rate to receive packets, and perform synchronization and CFO compensation. Unless otherwise stated, USRP nodes operate at 900 MHz, which is the frequency band specified in IEEE 802.11ah.

In Figure 9, we compare the decoding performance under different modulation schemes. The clock rate of T-Fi receiver is  $8\times$  of the standard 802.11 receiver. As expected, for all modulations demonstrated, T-Fi outperforms the standard 802.11 receiver. We also observe that T-Fi substantially reduces BER compared to the standard receiver for high-order modulations. When  $\text{SNR}=9$  dB, the BER of T-Fi is merely 25% of the standard receiver's BER. The reason is that high-order modulations are more error-prone under low SNR conditions, which leaves more room for overclocking to correct these errors.

Another observation is that the decoding performance of T-Fi for 64QAM (16QAM) is comparable to that of the

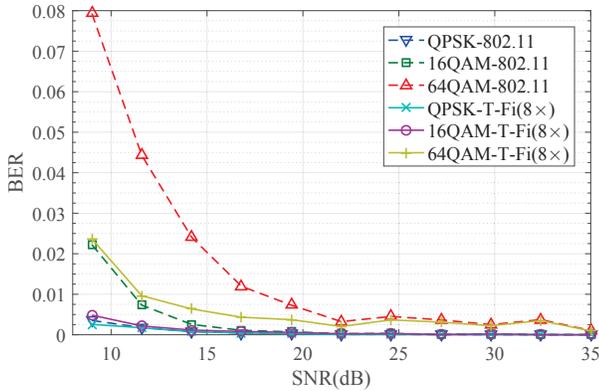


Fig. 9. BER for different modulations.

standard receiver for 16QAM (QPSK). This result implies that T-Fi can use higher-order modulations for packet transmission compared to standard Wi-Fi, and thus delivers roughly 4× data rate. Hence, we can envision a new dimension of the T-Fi is to improve the throughput in IoT Wi-Fi.

**Impact of wireless environment.** As analyzed in Section III, the decoding performance depends on channel response and noise. In our previous experiments, all nodes are in the same lab, and we focus on the decoding performance under different SNR conditions. In this experiment, we evaluate the impact of wireless environments, which lead to different multipath fading and shadowing. We repeat the previous set of experiments at three additional locations in the same building but with significantly different propagation environments. The locations of IoT nodes and the AP include a hallway in the lab, a corridor outside the lab, and a meeting room next to the lab, as illustrated in Figure 7. We use a fixed PGA to set the same TX power for all locations.

Figure 10(a) compares the decoding performance under different clock rates at different locations. For all locations, T-Fi outperforms the standard 802.11 decoding, and higher clock rate T-Fi receivers achieve lower BER than the lower clock rate ones. Compared to the standard receiver, T-Fi at 8× clock rate reduces BER to 56%-23% across all locations. T-Fi yields higher performance gain in the hallway and the corridor. The reason behind this is that the BER in these two locations falls in a sweet range where there is room for BER improvement while noise can be amortized by exploiting overclocking. The results confirm that T-Fi achieves stable performance gain over a wide range of wireless environment. Figure 10(b) further shows the BER performance for different modulation schemes at the four locations. Though the BER of different modulations varies significantly across different locations, T-Fi achieves steady BER gain over the standard receiver.

**Impact of carrier frequency.** Previous experiments use 900 MHz as the carrier frequency, which is the band specified by IEEE 802.11ah. Recently, several leading companies have been exploring the possibility of using 2.4 GHz for IoT trans-

missions. To meet this potential demands, we now evaluate the BER performance in both carrier frequencies under various SNR conditions.

Figure 11 shows the BER for 16QAM decoding. We have the following observations: (i) The BER performance at both frequencies has the same trend and is comparable in most cases; (ii) T-Fi achieves lower BER compared to the standard receiver at both frequencies. The results validate the feasibility of T-Fi at both carrier frequencies.

## VII. RELATED WORK

**IoT Wi-Fi standard.** IEEE 802.11ah, known as Wi-Fi HaLow [4], is the first and only Wi-Fi protocol dedicatedly designed for IoT. The target of this IoT Wi-Fi protocol is to utilize the Wi-Fi technology to provide low power and long range transmission for the emerging low-end IoT devices. It reuses the OFDM PHY as in IEEE 802.11ac, but provides a bundle of low power features, particularly at the MAC layer. Research efforts have focused on enhancing the MAC layer [12], [13]. Our goal is to design a transceiver architecture for IoT Wi-Fi networks that (i) completely conforms to the IoT Wi-Fi standard without any protocol modifications, and at the same time (ii) take full advantage of AP to further reduce the TX power of IoT devices. Our compliant design can be seamlessly integrated into the existing IoT Wi-Fi protocol. Existing Wi-Fi power models [14]–[19], [19]–[21] show that power amplifier operations dominate the total power consumption of Wi-Fi radios. Thus, the saved power budget by our design can facilitate lower power or longer range transmission, which fully aligns with the targets of IoT Wi-Fi protocols.

**Downclocking.** Recently, researchers have demonstrated that downclocking Wi-Fi radios can effectively reduce the energy consumption during packet reception and idle listening. E-Mili [7] pioneers this kind of mechanisms to downclock receiver’s clock rate during idle listening, and switches to full clock rate for packet reception. SloMo [22] enables packet decoding in IEEE 802.11b while downclocking by exploiting the sparsity in direct sequence spread spectrum (DSSS) PHY. SEER [8] designs a special preamble to allow narrowband receiver sensing a much wider band signal without boosting the sampling rate. Downclocking for packet reception has been extended to OFDM-based Wi-Fi by leveraging the gap between modulation and SNR [6]. Instead of exploiting PHY redundancy, Sampleless Wi-Fi [5], [23] utilizes the retransmission opportunities to decode packets at downclocked rates without relying on PHY redundancy, and thus makes downclocking OFDM packet reception feasible under low SNR conditions. T-Fi takes an opposite approach by overclocking the receiver’s radio to reduce the power consumption of the transmitter, which is complementary to downclocking approaches by bringing low power to the transmitter side.

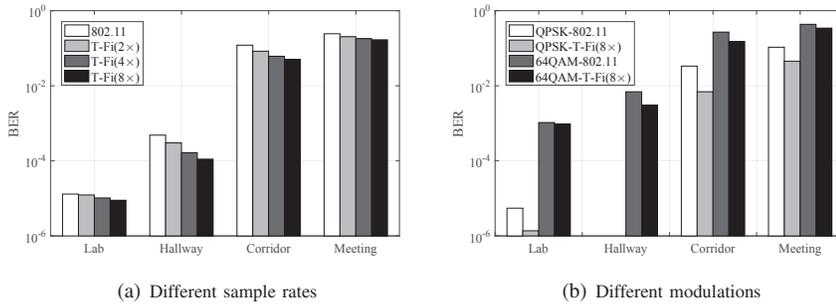


Fig. 10. BER in different locations.

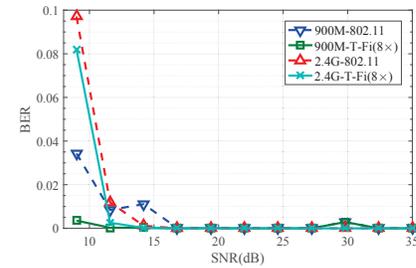


Fig. 11. BER under different carrier frequencies.

## VIII. CONCLUSION

This paper introduces T-Fi, an asymmetric transceiver paradigm for IoT that pushes power burden to the AP side, and enables IoT devices to transmit packets at power levels that are even lower than the minimal power required by conventional receivers. We think this is an important design point in IoT communications, and the teeter-totter fashion in T-Fi has significant ramifications in the new transceiver design for IoT Wi-Fi protocols. Our experimental evaluation confirms the benefits of T-Fi in real environments. We hope the design can contribute the wireless community by providing some insights for future transceiver design that takes advantage of the hardware asymmetry between APs and IoT devices.

## ACKNOWLEDGMENT

The research was supported in part by the National Science Foundation of China under Grant 61502114, 91738202, 61729101, and 61531011, Major Program of National Natural Science Foundation of Hubei in China with Grant 2016C-FA009, and the Fundamental Research Funds for the Central Universities with Grant number 2015ZDTD012.

## REFERENCES

- [1] "Android wear help: Connect to Wi-Fi." [Online]. Available: <http://support.google.com/androidwear>
- [2] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, pp. 1–11, 2011.
- [3] W. Wang, Y. Chen, Q. Zhang, K. Wu, and J. Zhang, "Less transmissions, more throughput: Bringing carpool to public wlangs," *IEEE Trans. on Mobile Comput.*, vol. 15, no. 5, pp. 1168–1181, 2016.
- [4] *Wi-Fi HaLow: Low Power, Long Range Wi-Fi*, Wi-Fi Alliance, <http://www.wi-fi.org/discover-wi-fi/wi-fi-halow>.
- [5] W. Wang, Y. Chen, L. Wang, and Q. Zhang, "Sampleless wi-fi: Bringing low power to wi-fi communications," *IEEE/ACM Trans. Netw.*, 2017.
- [6] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren, "Enfold: downclocking OFDM in WiFi," in *Proc. ACM MobiCom*, 2014.
- [7] X. Zhang and K. G. Shin, "E-mili: energy-minimizing idle listening in wireless networks," in *Proc. ACM MobiCom*, 2011.
- [8] W. Wang, Y. Chen, Z. Wang, J. Zhang, K. Wu, and Q. Zhang, "Wideband spectrum adaptation without coordination," *IEEE Trans. on Mobile Comput.*, vol. 16, no. 1, pp. 243–256, 2017.
- [9] "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) Specifications, journal=IEEE Std. 802.11, year=2007,."
- [10] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on IEEE 802.11 ah: An enabling networking technology for smart cities," *Elsevier Computer Communications*, vol. 58, pp. 53–69, 2015.
- [11] *MAX2831/MAX2832 World-Band Transceiver ICs Datasheets*, Maxim Integrated, <https://datasheets.maximintegrated.com/en/ds/MAX2831-MAX2832.pdf>.
- [12] C. W. Park, D. Hwang, and T.-J. Lee, "Enhancement of IEEE 802.11 ah MAC for M2M communications," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1151–1154, 2014.
- [13] R. P. Liu, G. J. Sutton, and I. B. Collings, "Power save with offset listen interval for IEEE 802.11 ah smart grid communications," in *Proc. IEEE ICC*, 2013, pp. 4488–4492.
- [14] D. Qiao, S. Choi, A. Jain, and K. G. Shin, "Miser: an optimal low-energy transmission strategy for IEEE 802.11 a/h," in *Proc. ACM MobiCom*, 2003, pp. 161–175.
- [15] E.-S. Jung and N. H. Vaidya, "An energy efficient mac protocol for wireless LANs," in *Proc. IEEE INFOCOM*, vol. 3, 2002, pp. 1756–1764.
- [16] V. Baiamonte and C.-F. Chiasserini, "Saving energy during channel contention in 802.11 WLANs," *Springer Mobile Networks and Applications*, vol. 11, no. 2, pp. 287–296, 2006.
- [17] "Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs, author=Bruno, Raffaele and Conti, Marco and Gregori, Enrico, journal=IEEE Transactions on Mobile Computing, volume=99, number=1, pages=10–31, year=2002,."
- [18] M. M. Carvalho, C. B. Margi, K. Obraczka, and J. Garcia-Luna-Aceves, "Modeling energy consumption in single-hop IEEE 802.11 ad hoc networks," in *Proc. IEEE ICCCN*, 2004, pp. 367–372.
- [19] J.-C. Chen and K.-W. Cheng, "Edca/ca: Enhancement of IEEE 802.11 e EDCA by contention adaption for energy efficiency," *IEEE Transactions on Wireless Communications*, vol. 7, no. 8, 2008.
- [20] M. Ergen and P. Varaiya, "Decomposition of energy consumption in IEEE 802.11," in *Proc. IEEE ICC*, 2007, pp. 403–408.
- [21] A. Garcia-Saavedra, P. Serrano, A. Banchs, and M. Hollick, "Energy-efficient fair channel access for IEEE 802.11 WLANs," in *Proc. of IEEE WoWMoM*, 2011, pp. 1–9.
- [22] F. Lu, G. M. Voelker, and A. C. Snoeren, "Slomo: Downclocking WiFi communication," in *USENIX NSDI*, 2013.
- [23] W. Wang, Y. Chen, L. Wang, and Q. Zhang, "From rateless to sampleless: Wi-Fi connectivity made energy efficient," in *Proc. IEEE INFOCOM*, 2016, pp. 1–9.