LiteNap: Downclocking LoRa Reception

Xianjin Xia, *Member, IEEE, ACM*, Yuanqing Zheng, *Member, IEEE, ACM*, and Tao Gu, *Senior Member, IEEE, Member, ACM*

***Abstract*— This paper presents LiteNap which improves the energy efficiency of LoRa by enabling LoRa nodes to operate in a downclocked 'light sleep' mode for packet reception. A fundamental limit that prevents radio downclocking is the Nyquist sampling theorem which demands the clock-rate being at least twice the bandwidth of LoRa chirps. Our study reveals under-sampled LoRa chirps suffer frequency aliasing and cause ambiguity in symbol demodulation. LiteNap addresses the problem by leveraging an empirical observation that the hardware of LoRa radio can cause phase jitters on modulated chirps, which result in frequency leakage in the time domain. The timing information of phase jitters and frequency leakages can serve as physical fingerprints to uniquely identify modulated chirps. We propose a scheme to reliably extract the fingerprints from under-sampled chirps and resolve ambiguities in symbol demodulation. We update the reception pipeline of LoRa radio to enable reliable packet detection and decoding when operating in downclocked mode. We implement LiteNap on a software defined radio platform and conduct trace-driven evaluation to validate the proposed strategies. Experiment results show that LiteNap can downclock LoRa receiver to sub-Nyquist rates for energy savings (*e.g.*, 1/8 of Nyquist rate), without substantially affecting packet reception performance (*e.g.*, >95% packet reception rate).**

***Index Terms*— Internet of Things, LoRaWAN, packet reception, power saving, downclocking.**

I. Introduction

THE recent advances of LP-WANs (*i.e.*, Low Power Wide Area Networks) enable wireless network access for long-term operated IoT devices. Among many LP-WAN technologies (*e.g.*, SigFox [1], NB-IoT [2]), LoRaWAN has received wide attention from industry and academia due to its capability in terms of long range, low power, *etc* [3]. The PHY layer of LoRaWAN adopts Chirp Spread Spectrum (CSS) modulation, which is robust against noise, multi-path and Doppler effects.

Despite the excellent communication performance of LoRa, energy efficiency remains a major concern since LoRa nodes are typically battery-powered and expected to operate over a long time (*e.g.*, $\geq$ 10 years) without replacing the battery. However, measurement study [4] reports that in practice the lifetime of LoRa nodes can be much shorter in case of frequent packet transmission and reception [5]. As the on-air time of LoRa packet is much longer than that of other wireless technologies (*e.g.*, WiFi, ZigBee) [6], [7], LoRa nodes must stay awake for a longer time to send and receive a packet during packet transmission. As a result, the per-packet power consumption of LoRa can be much higher than those of conventional radios. The most power-hungry components of LoRa radio are Analog-to-Digital Converter (*i.e.*, ADC) and the processing unit (*i.e.*, MCU).

Orthogonal to the conventional energy saving schemes such as duty-cycling, our work explores an alternative approach to reduce the per-packet power consumption of LoRa using downclocking. The power consumption of MCU and ADC is generally proportional to the operating clock rates [8]. By decreasing the clock rates, we expect to proportionally reduce the power consumption of LoRa radio.

However, it is challenging to downclock LoRa radio to reduce power consumption without affecting communication performance. It is well known that the clock-rate of ADC is fundamentally limited by the Nyquist's theorem, which requires the sampling rate to be at least twice the signal bandwidth. If the clock-rate of a receiver falls below the Nyquist rate (*i.e.*, twice the maximum frequency of LoRa chirps), the frequency aliasing may lead to incorrect demodulation. As LoRa spreads the chirp frequency across the entire bandwidth, it leaves little room for downclocking LoRa receiver without causing frequency aliasing. Aiming at reducing the power consumption of LoRa receiver, this paper asks a question: *can we decode LoRa packets sampled at sub-Nyquist rate?*

We conduct extensive measurement study and theoretical analysis on LoRa packet reception at sub-Nyquist rates. Our study yields two observations: (1) The frequency aliasing may fold two LoRa chirps separated by the sampling frequency into the same aliased frequency. As a result, the two LoRa chirps measured at a sub-Nyquist rate may resemble each other and cause ambiguity in demodulation. Hence, *the key to demodulate under-sampled chirps lies in how to resolve the ambiguity caused by frequency aliasing due to under-sampling.* (2) As a LoRa chirp continuously sweeps across a pre-configured LoRa band, *frequency leakage* inevitably occurs when the instantaneous frequency suddenly changes from its maximum to minimum. Since LoRa varies the starting frequency of chirp to encode different data and the increasing rate of frequency remains constant, frequency leakage happens at distinct time in different LoRa chirps. Such frequency

Manuscript received April 21, 2020; revised April 4, 2021; accepted July 1, 2021; approved by IEEE/ACM Transactions on Networking Editor E. Uysal. Date of publication July 21, 2021; date of current version December 17, 2021. This work was supported in part by the Hong Kong General Research Fund (GRF) under Grant PolyU 152165/19E, in part by the Start-up Fund for Research Assistant Professors (RAPs) under the Strategic Hiring Scheme of Hong Kong PolyU under Grant P0036217, and in part by the Australian Research Council (ARC) Discovery Project under Grant DP190101888. *(Corresponding author: Yuanqing Zheng.)*

Xianjin Xia and Yuanqing Zheng are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csxxia@comp.polyu.edu.hk; csyqzheng@comp.polyu.edu.hk).

Tao Gu was with the School of Computer Science and IT, RMIT University, Melbourne, VIC 3000, Australia. He is now with the Department of Computing, Macquarie University, Sydney, NSW 2109, Australia (e-mail: tao.gu@mq.edu.au).

Digital Object Identifier 10.1109/TNET.2021.3096990

leakage in the time domain can thus be used as a *physical fingerprint* to uniquely identify a LoRa chirp and help resolve the ambiguity caused by frequency aliasing. More importantly, as the frequency leakage spans across the full frequency band, the timing information can still be reserved even sampled at sub-Nyquist rates.

Based on the above observations, we propose a decoding method named LiteNap for down-clocked LoRa reception. We first exploit the prior knowledge of LoRa preamble to detect chirp boundaries. Then, we demodulate under-sampled chirps and measure the initial frequency of LoRa chirps with frequency aliasing. We finally disambiguate the aliasing with the timing information of frequency leakages. Intuitively, we use the timing information of frequency leakage to determine the missing bits caused by under-sampling.

It entails tremendous challenges to extract timing information of frequency leakage from under-sampled chirps. Firstly, as frequency leakage occurs in a rather short time, a downclocked radio with low sampling rates may miss detecting frequency leakage. Secondly, as the power strength of frequency leakage is weak, which can be buried below noise, it is non-trivial to reliably detect from the frequency domain. Our study reveals that frequency leakages are essentially caused by the phase jitters of chirp samples, which are introduced by the hardware of LoRa modem. We find that the hardware phase jitters will add constant phase shifts to all modulated samples transmitted after the jitters. We leverage the finding to extract symbol fingerprint in the phase domain. Specifically, we check the phase of received chirp samples to detect phase shift and extract the timing information as fingerprint. We experimentally demonstrate that the phase-based approach can correctly extract unique fingerprints from under-sampled chirps in various SNR conditions.

A practical challenge is how to reliably detect symbol timing (*i.e.*, chirp boundaries) from under-sampled preamble. The conventional method of correlating preamble with known base-chirp detects incorrect symbol edges due to frequency aliasing of preamble chirps. The relationship of correlation becomes weak in case of under-sampling. Our solution leverages the finding that the fingerprints of chirps in preamble (*i.e.*, phase jitters) appear at around chirp boundaries and can be detected reliably from undersampled chirps in the phase domain. As a LoRa preamble consists of repetitive base-chirps with identical fingerprint, the detected phase jitters appear uniformly in time. We design an algorithm to extract the correct symbol timing from the detected timing of phase jitters.

LiteNap introduces a downclocked operation mode for LoRa radio and seamlessly integrates into the LoRaWAN protocol. It adds two new states into the existing state transition machine of LoRaWAN that schedules LoRa radios to detect and receive packets at downclocked rate. LiteNap dynamically adapts the downclocking factor to channel conditions to achieve a balance between power-saving and reliability of packet reception. We implement and evaluate LiteNap on a Software Defined Radio platform. Evaluation results show that LiteNap can reduce half the energy consumption by downclocking a receiver radio to 1/8 of the Nyquist-rate while achieving high packet decoding accuracy ($> 95\%$).

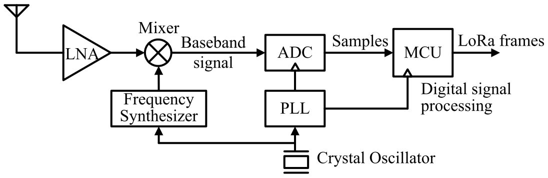


Fig. 1. Simplified receiver chain of LoRa node [9].

In summary, this paper makes the following contributions.

- We explore the feasibility of downclocking a LoRa radio to sub-Nyquist rate for power saving without impairing the basic communication functionality.
- We reveal the specific aliasing structure of under-sampled LoRa chirps and discover a physical fingerprint that can help recover symbol from under-sampled chirps.
- We propose LiteNap that enables LoRa radio to operate in downclocked mode to detect, receive and decode packets. LiteNap allows LoRa radio to adapt clock-rate to packet reception and power saving performance.
- We implement a prototype of LiteNap on software defined radio platform and validate the proposed strategies through testbed experiments and trace-driven simulations.

II. Background and Motivation

A. LoRa Modulation

The PHY modulation of LoRa adopts Chirp Spread Spectrum (CSS), where the frequency of a LoRa chirp increases linearly with time and sweeps through a predefined bandwidth. A *base chirp* is represented as $C(t) = e^{j2\pi(\alpha t + f_0)t}$, where f_0 and $(\alpha t + f_0)$ denote the initial frequency and instantaneous frequency at time t, respectively. The frequency increasing rate (α) and time duration (T_{chirp}) of a chirp is determined by two parameters of CSS modulation: spreading factor (SF) and frequency bandwidth (BW), *i.e.*, $T_{chirp} = \frac{2^{SF}}{BW}$, $\alpha = \frac{BW}{T_{chirp}}$.

In modulation, LoRa varies the initial frequency of chirp signal to represent different data. The modulation procedure can be represented as follows.

$$S(t, f_{sym}) = C(t) \cdot e^{j2\pi f_{sym} t} \tag{1}$$

where f_{sym} is the starting frequency of chirp $S(t, f_{sym})$ and $C(t)$ represents a base chirp. We see that the starting frequency f_{sym} carries the information of modulated data. To demodulate a received symbol from a chirp $S(t, f_{sym})$, we multiply the received chirp with the conjugate of a base chirp (denoted as $C^{-1}(t)$) as below:

$$S(t, f_{sym}) \cdot C^{-1}(t) = e^{j2\pi f_{sym} t}. \tag{2}$$

Then, we perform FFT (*i.e.*, Fast Fourier Transform) to derive the starting frequency (*i.e.*, f_{sym}) and demodulate the chirp.

B. Radio Power vs. Clock Rate

Fig.1 shows the receiver chain of a LoRa radio [9]. The RF frontend first down-converts the received signal to intermediate frequency using a mixer. Next, an Analog-to-Digital

TABLE I

POWER PROFILES OF A LORA NODE IN VARIOUS OPERATION MODES

Modes		Enabled Components	Current Draw
Active	Transmit	MCU + TX Chains	20∼120 mA*
	Receive	MCU + RX Chains	11.5 mA
	Standby	MCU (disabled RF & PLL)	1.6∼1.8 mA
Sleep		Circuit	0.2∼1.5 μA

*The current draw varies with respect to the transmission power of 7∼20 dBm.

Converter (ADC) samples the baseband signal, and passes the samples to a MCU for further processing. The radio is typically driven by a 32 MHz crystal oscillator, which feeds both the frequency synthesizer and Phase-Locked-Loop (PLL). The output of PLL controls the sampling rate of ADC and the clock-rate of MCU.

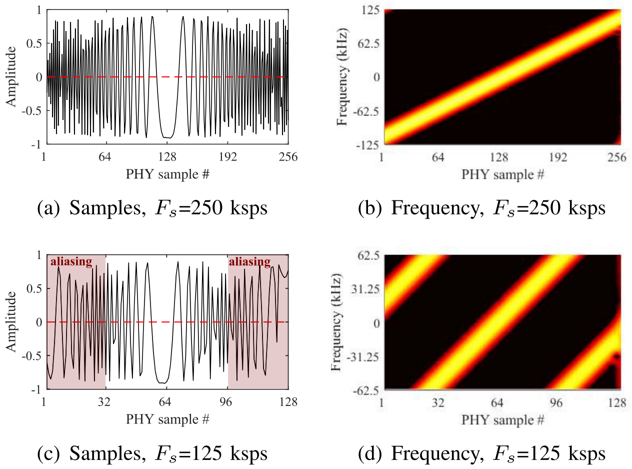
When a LoRa radio is awake in scheduled TX/RX windows, it operates in one of the three modes, *i.e.*, Transmit, Receive or Standby. Table I compares the power profiles of various operation modes for a LoRa node based on the Semtech SX1276 datasheet [9]. When a radio is in the Receive mode, ADC continuously samples the channel to detect, receive and decode incoming packets. As such, ADC and MCU are the most power-hungry components in the Receive mode. When a radio turns to Standby, most components on the receiver chain (*e.g.*, ADC) except MCU are powered off [4], [9]. The power consumption of MCU and ADC generally follows $P \propto V^2\ f$ [8], [10], where V is the supply voltage, f denotes the clock-rate for MCU and the sampling rate for ADC, respectively.

We aim to reduce the active power consumption of LoRa radio by downclocking the power-hungry components. This can be realized by changing the PLL configuration of desired output frequency, which is programmable in practice. As a LoRa node typically stays in Standby for most of the time [4], we expect significant reduction in power consumption if the clock-rate of MCU can be decreased in the Standby mode.

However, a key problem arises: When a radio changes the PLL frequency to downclock MCU, the sampling rate of ADC also decreases, since the two components share the same clock as illustrated in Fig.1. The sampling rate of ADC is restricted fundamentally by the Nyquist theorem. If the sampling rate falls below the Nyquist rate (*i.e.*, twice the signal bandwidth), a LoRa node may not be able to demodulate incoming LoRa chirps because of frequency aliasing. Our work tackles the problem by exploring feasible approaches to decode LoRa packets at sub-Nyquist sampling-rates.

C. Problem Statement

In this paper, we target at energy saving for LoRa nodes. As the air-time of a LoRa packet is relatively long (*e.g.*, 100∼1000 ms), LoRa radio has to spend long time and energy consumption to detect and receive a packet. We aim to reduce the operational power of LoRa radio through downclocking. In particular, we focus on reducing the power consumption of MCU and ADC when LoRa radio stays in Receive and Standby modes that account for nearly 50% of the total power consumption of LoRa node [4]. We operate LoRa radio in a downclocked or 'light sleep' mode for packet detecting, receiving and decoding, and switch back to normal clock-rate for packet sending. As downclocking may impair the SNR (*i.e.*, Signal-to-Noise-Ratio) of received signal, *how to recover packet from under-sampled signals while ensuring reliability is key of the problem.*



(a) Samples, F_s=250 ksps (b) Frequency, F_s=250 ksps

(c) Samples, F_s=125 ksps (d) Frequency, F_s=125 ksps

Fig. 2. The time-domain and frequency-domain presentations of a LoRa chirp when sampling at Nyquist-rate (a,b) and sub-Nyquist rate (c,d).

III. DEMODULATION OF UNDER-SAMPLED CHIRPS

A. Downclocked Reception Model

In this section, we investigate the frequency aliasing effect of LoRa chirps when sampling at sub-Nyquist rates. Consider the transmitted chirp of a single symbol, *i.e.*, $S(t, f_{sym})$. We denote the received signal as

$$R(t, f_{sym}) = h(t)e^{j2\pi\Delta f_{cfo}t} \cdot S(t, f_{sym}) + n(t),\ \ t \in [0, T_{chirp})$$

where, $n(t)$ is noise, $h(t)$ denotes the changes of amplitude and phase induced by wireless channel, and Δf_{cfo} denotes the carrier frequency offset between transmitter and receiver.

The received signal (*i.e.*, $R(t, f_{sym})$) is sampled by ADC. A normal receiver shall sample the chirp above Nyquist rate, which is $F_s = BW$ as chirp frequency varies within $-\frac{BW}{2} \sim \frac{BW}{2}$. We denote the discrete signal samples with

$$R(k) = R(\frac{k}{BW}, f_{sym}), \quad k = 0, 1, \ldots, 2^{SF} - 1.$$

If the receiver reduces the sampling rate by a factor of D (*i.e.*, with a sampling-rate $F_s = \frac{BW}{D}$), the signal samples become

$$R(Dk) = R(\frac{Dk}{BW} + \Delta t, f_{sym}), \quad k = 0, 1, \ldots, \lfloor \frac{2^{SF}}{D} \rfloor - 1. \tag{3}$$

where, Δt represents the time offset between the first sample and the arrival time of the chirp (*i.e.*, chirp boundary).

To illustrate how frequency aliasing affects LoRa demodulation, we use a USRP to receive LoRa packets transmitted by a Semtech SX1276 based LoRa node with $SF = 8$ and $BW = 250$ kHz. The sampling rate of USRP is 250 ksps (*i.e.*, the Nyquist rate). We emulate a down-sampling factor of D by drawing one sample from every D samples. Fig.2 shows the time-domain and frequency-domain results of the

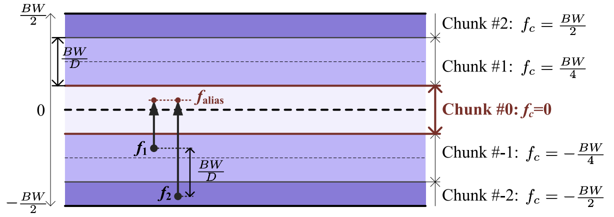

$\frac{BW}{2}$
$\frac{BW}{D}$
0
f_{alias}
f_1
f_2
$\frac{BW}{D}$
$-\frac{BW}{2}$
Chunk #2: $f_c = \frac{BW}{2}$
Chunk #1: $f_c = \frac{BW}{4}$
Chunk #0: f_c=0
Chunk #-1: $f_c = -\frac{BW}{4}$
Chunk #-2: $f_c = -\frac{BW}{2}$

Fig. 3. Frequency aliasing model (down-clocking factor $D = 4$): f_1, f_2 are folded to the same aliased frequency (f_{alias}), resulting in ambiguity.

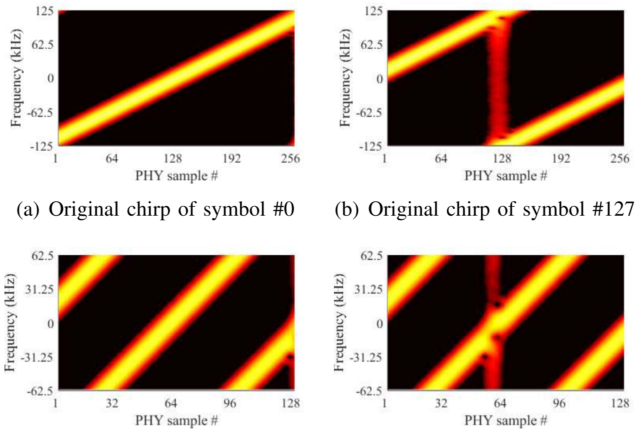
same LoRa chirp under different sampling-rates. In Fig.2(c), we see that as the sampling-rate decreases below the Nyquist rate, the obtained signal samples become sparse. As expected, the frequency of the signal reconstructed from sub-Nyquist samples are distorted from that of the original signal, especially for the high-frequency components (see Fig.2(b,d)).

According to the Nyquist-Shannon sampling theorem, when sampling at the rate of F_s, we can only reconstruct the signal of frequency ranging from $-\frac{F_s}{2}$ to $\frac{F_s}{2}$. If the receiver radio samples LoRa chirps at a sub-Nyquist rate $F_s = \frac{BW}{D}$, the original frequency of chirp signal which spreads across the entire LoRa bandwidth (*i.e.*, $[-\frac{BW}{2}, \frac{BW}{2}]$) will be folded into $[-\frac{BW}{2D}, \frac{BW}{2D}]$. Specifically, we can characterize the frequency aliasing of LoRa chirps as follows.

As illustrated in Fig.3, we divide the entire frequency band into $D+1$ chunks. We represent each chunk as $[f_c - \frac{BW}{2D}, f_c + \frac{BW}{2D})$, where $f_c = n_c\frac{BW}{D}$ denotes the central frequency of chunk n_c, and n_c denotes the chunk ID. Because of the frequency aliasing effect, a frequency $f = n_c\frac{BW}{D} + f_{alias}$ of outer chunks (*i.e.*, $|n_c| > 0$) will be folded into f_{alias} of chunk 0, where $f_{alias} \in [-\frac{BW}{2D}, \frac{BW}{2D})$.

The frequency aliasing effect of downsampling can cause *symbol ambiguity problem* in LoRa demodulation. As shown in Fig.4, the chirps of symbol #0 ($f_{sym0} = -125$kHz) and #127 ($f_{sym127} = 0$kHz), when sampled at half the Nyquist rate, will be folded into the same chunk. More importantly, the starting frequencies of both aliased symbols become the same (*i.e.*, 0kHz). As a result, we cannot distinguish the two aliased symbols by examining their starting frequencies. Generally, with a downsampling factor of D, such ambiguity would happen between the two symbols f_1 and f_2, if $|f_1 - f_2| = n\frac{BW}{D},\ n = 1, 2, \ldots, D-1$.

Therefore, the key problem in downclocked LoRa reception is *how to resolve the ambiguity of under-sampled chirps and correctly demodulate an aliased chirp*. We observe from Fig.4 that although the starting frequencies of aliased symbols are the same, comparing Fig.4 (c,d), we see that the two aliased symbols exhibit different patterns. In particular, while there is a vertical frequency leakage band appearing at PHY sample 64 in Fig.4 (d), there is no such a vertical band at the location in Fig.4 (c). As a matter of fact, the vertical band appears at PHY sample 128 in Fig.4 (c). That is because when a chirp frequency suddenly jumps from $\frac{BW}{2}$ to $-\frac{BW}{2}$, there is frequency leakage spanning the entire bandwidth. Comparing Fig.4 (b,d), we also observe that the timing information of frequency leakage is reserved even in the aliased chirp. It motivates us to exploit the timing information of frequency leakage to resolve symbol ambiguity.



(a) Original chirp of symbol #0 (b) Original chirp of symbol #127

(c) Aliased chirp of symbol #0 (d) Aliased chirp of symbol #127

Fig. 4. (a,b) The original chirps of symbol #0 and #127 (sampling above the Nyquist-rate); (c,d) The reconstructed signals of symbol #0 and #127 when sampling at half the Nyquist-rate.

B. Demodulating Aliased Symbol

We adapt the standard method of Eq.(2) to demodulate an under-sampled LoRa chirp (*i.e.*, Eq. (3)). Specifically, we replace $C^{-1}(t)$ in Eq.(2) with a discrete under-sampled down-chirp, denoted by $C^{-1}(\frac{Dk}{BW}),\ k = 0, 1, \ldots, \lfloor\frac{2^{SF}}{D}\rfloor - 1$. The operation is represented as follows.

$$\begin{aligned}R(Dk)\cdot C^{-1}(\frac{Dk}{BW}) &= R(\frac{Dk}{BW} + \Delta t, f_{sym})\cdot C^{-1}(\frac{Dk}{BW})\\ &\approx h(\frac{Dk}{BW} + \Delta t)e^{j2\pi\Delta f_{cfo}(\frac{Dk}{BW}+\Delta t)}\\ &\quad\cdot S(\frac{Dk}{BW} + \Delta t, f_{sym})\cdot C^{-1}(\frac{Dk}{BW})\end{aligned}$$

We omit noise $n(t)$ due to the fact that the signal strength, while being added up across the whole chirp, is usually much higher than noise. Note that the time offset of under-sampling (Δt) can transform to a frequency offset of $\Delta f_{sample} = \frac{\Delta t}{2^{SF}}$ in LoRa chirp. We have $S(\frac{Dk}{BW} + \Delta t, f_{sym}) = S(\frac{Dk}{BW}, f_{sym} + \Delta f_{sample})$. Substituting into the above equations, we get

$$\begin{aligned}&R(Dk)\cdot C^{-1}(\frac{Dk}{BW})\\ &\quad\approx h(\frac{Dk}{BW} + \Delta t)e^{j2\pi\Delta f_{cfo}(\frac{Dk}{BW}+\Delta t)}\\ &\quad\quad\cdot e^{j2\pi(f_{sym}+\Delta f_{sample})\cdot\frac{Dk}{BW}}\\ &\quad\approx e^{j2\pi(f_{sym}+\Delta f_{sample}+\Delta f_{cfo})\cdot\frac{Dk}{BW}}\end{aligned} \tag{4}$$

We exclude $h(t)$ from the equations because $h(t)$ can be measured and cancelled by leveraging the prior knowledge of chirps in preamble. Next, we expect to perform FFT on $R(Dk)\cdot C^{-1}(\frac{Dk}{BW})$ to derive the encoded symbol (*i.e.*, f_{sym}). However, two issues still need to be handled:

1) Frequency Offset Cancellation: To demodulate the correct symbol (*i.e.*, f_{sym}), we must remove $(\Delta f_{sample} + \Delta f_{cfo})$ from Eq. (4). As the frequency offsets of radio hardware (Δf_{cfo}) and sample timing (Δf_{sample}) remain relatively stable across all chirps in the same packet, we can detect $(\Delta f_{sample} + \Delta f_{cfo})$ from the preamble of LoRa packet (by leveraging the fact that $f_{sym} = 0$ in preamble) and subtract

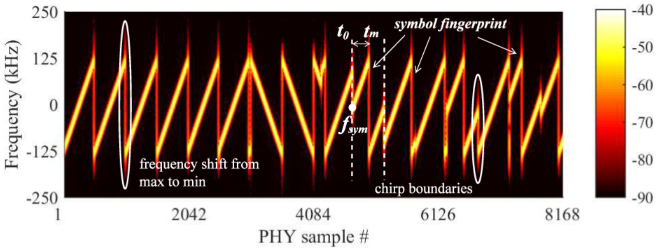


Fig. 5. Illustration of frequency leakage in the received LoRa chirps.

$(\Delta f_{sample} + \Delta f_{cfo})$ from the FFT results of symbols in the payload. We present the detailed method in Section IV.

2) Ambiguity Mitigation: As the chirp signal is received below the Nyquist sampling-rate, by performing FFT on $R(Dk) \cdot C^{-1}(\frac{Dk}{BW})$, we can only obtain an aliased frequency f_{alias}. According to the aliasing model, the real symbol can be any of $f_{alias} + n\frac{BW}{D},\ n = 0, 1, \ldots, D-1$. We resolve the ambiguity by exploiting additional information.

Before moving on to ambiguity mitigation, we analytically study the impacts of down-sampling on symbol demodulation. We focus on the Signal-to-Noise-Ratio (SNR) of the FFT response of Eq. (4). For clarity, we use $x(n), n = 0, 1, \ldots, N-1$, to denote the time-domain samples of dechirped signal (*e.g.*, $R(Dk) \cdot C^{-1}(\frac{Dk}{BW})$). Suppose there is a single tone in $x(n)$. By performing FFT on $x(n)$, we obtain results as below.

$$X(f) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N} n \cdot f}, \quad f = 0, 1, \ldots, N-1. \tag{5}$$

Let M denote the amplitude of frequency response in $X(f)$ for the demodulated symbol f_{sym}. The signal power is $P_s = M^2$. The noise power P_n is the sum of power of all frequencies in $X(f)$ other than f_{sym}. Hence, the SNR of symbol demodulation at normal sampling rate is $\text{SNR}_0 = \frac{M^2}{P_n}$.

When we downsample by a factor of D, the number of samples contributing to $X(f)$ of Eq. (5) decreases to $\frac{N}{D}$. The amplitude of frequency response of f_{sym} reduces to $\frac{M}{D}$, and the signal power becomes $P_s = \frac{M^2}{D^2}$. Note that noise power is proportional to the bandwidth of frequency in $X(f)$ which shrinks to $\frac{1}{D}$ of the normal band, the resulting noise power is $\frac{P_n}{D}$. Therefore, the SNR changes to the following.

$$\text{SNR}_D = \frac{M^2/D^2}{P_n/D} = \frac{\text{SNR}_0}{D}. \tag{6}$$

The take-away point is that *we would suffer 3 dB power loss whenever we cut the clock-rate by half (i.e., $D = 2$).*

C. Frequency-Based Ambiguity Mitigation

In this subsection, we propose a frequency-based approach to detect frequency leakage and mitigate the symbol ambiguities caused by frequency aliasing.

Symbol fingerprint. Fig.5 presents the spectrogram of a portion of a LoRa packet sent by a COTS LoRa node. We see from Fig.5 that there are frequency leakages from the main frequency of chirp signal (which indicates the instability of chirp frequency) as LoRa chirps experience sudden changes of frequency (*e.g.*, when a chirp shifts from the maximum frequency to the minimum or at chirp boundaries). The frequency leakage spans across the whole bandwidth, as displayed in Fig.5. We observe the same phenomenon in various types of COTS LoRa nodes (*e.g.*, Adafruit Feather 32u4, Dragino LoRa shield, *etc*.).

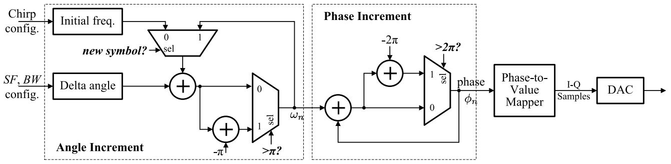


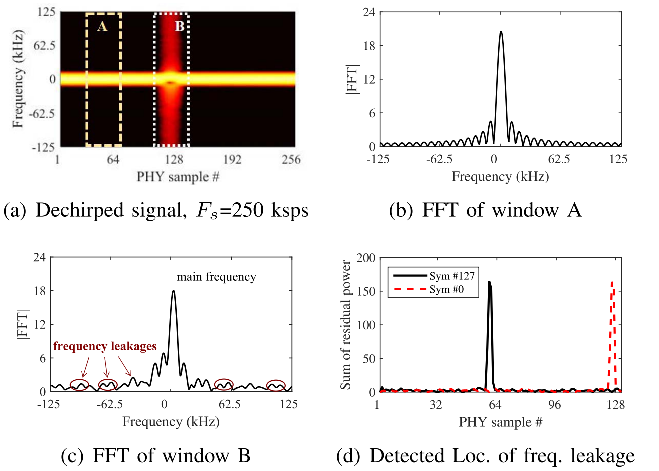
Fig. 6. Hardware implementation of CSS based on angle modulation.

Generally, the manufacture imperfection of radio electronics (*e.g.*, oscillator) can add random phase jitters to the modulated signal [11]. Based on the study of hardware implementations of CSS [12], [13] and Semtech datasheet [9], [14], we know that LoRa radio employs angle modulation (*i.e.*, phase-based modulation) to generate CSS samples. We present the hardware architecture of CSS modem in Fig.6. To handle the frequency shifts when a new symbol starts and when a chirp reaches the maximum frequency, the modem requires switching to a new modulation parameter or enabling additional blocks to amend the modulated phases. During such changes, LoRa radio will experience hardware instabilities that lead to phase jitters. The phase jitters of modulated signal would consequently exhibit as frequency leakage as shown in Fig.5.

The frequency leakage appearing around the time when a chirp shifts from the maximum frequency to the minimum can serve as a PHY fingerprint to uniquely identify an aliased symbol. As illustrated in Fig.5, let t_0 denote the starting time of the chirp, t_m the time of chirp frequency shifting from $\frac{BW}{2}$ to $-\frac{BW}{2}$. Since chirp frequency increases linearly with time, we can infer the starting frequency of chirp as $f_{sym} = \frac{BW}{2} - \alpha \cdot (t_m - t_0)$. Therefore, we can detect frequency leakage within a chirp and extract the timing information to resolve the ambiguity caused by frequency aliasing.

Fingerprint extraction. We detect the frequency leakage as follows. As in the standard demodulation process, we first perform a dechirp operation by multiplying the received chirp with $C^{-1}(t)$ (see Eq. (2)), which accumulates power into a main frequency (*i.e.*, f_{sym}). Next, we perform FFT on the results to detect if there is frequency leakage spanning across the whole bandwidth. Fig.7(a) presents the dechirped results of the chirp shown in Fig.4(b). We take two different segments (with/without frequency leakage) of the dechirped signal with the same segment length. We compare the FFT of two segments in Fig.7(b) and (c). As displayed in Fig.7(c), the irregular tiny spikes apart from the main peak indicate frequency leakage, yet the spikes in Fig.7(b) correspond to the sinc side lobes of the main frequency, which are introduced by FFT. To remove the influence of main frequency, we subtract the FFT result of Fig.7(b) from that of Fig.7(c), then sum up the residual power in all FFT bins to get the power of frequency leakage.

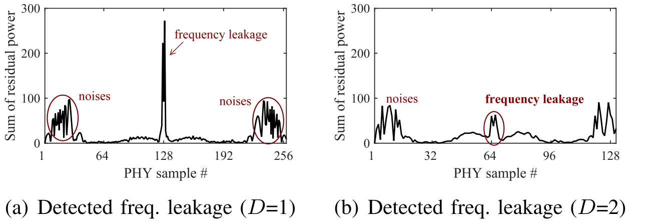
The timing of frequency leakage can be detected with a sliding-window method: We move an FFT window across



(a) Dechirped signal, F_s=250 ksps (b) FFT of window A

(c) FFT of window B (d) Detected Loc. of freq. leakage

Fig. 7. Frequency-based fingerprint detection: (a) Dechirping the received chirp signal of Fig.4(b); (b,c) FFT results of the signal in window A and B; and (d) Frequency leakages detected from the under-sampled chirps of symbol #0 and #127, ambiguity symbols have different fingerprints.



(a) Detected freq. leakage (D=1) (b) Detected freq. leakage (D=2)

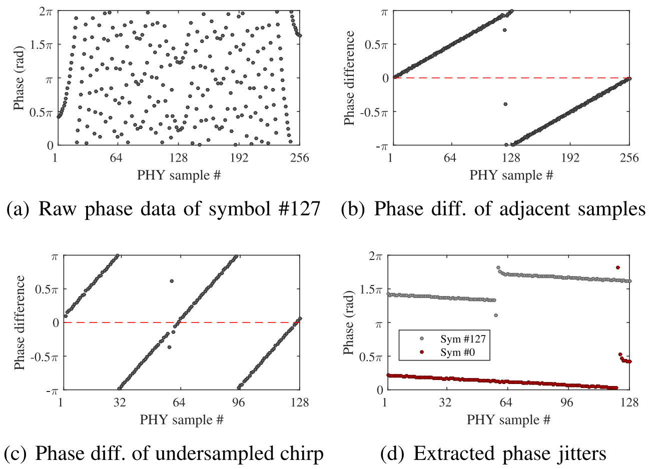
Fig. 8. Impacts of noise on the detection of frequency leakage: Detecting results of (a) fully-sampled and (b) under-sampled chirps. Frequency leakage is buried within noise in the case of down-sampling (*e.g.*, $D = 2$).

all samples of the chirp, and extract the power of frequency leakage in each window by subtracting the FFT of adjacent windows. Fig.7(d) presents the detected power of frequency leakage at different offsets of the chirp in Fig.7(a). We can adopt a threshold to detect the time of frequency leakage (*i.e.*, t_m). For the example shown in Fig.4, we can detect different t_m from the chirps of symbol #0 and #127, as displayed in Fig.7(d). We exploit the detected fingerprint information to resolve ambiguity of the two symbols.

Remarks. The frequency-based approach relies on detecting the power of frequency leakage. However, as frequency leakage appears in short time, it can be missed under lower sampling-rates. In addition, the method of power detecting is vulnerable to noise. Fig.8 presents the detecting results on a noisy chirp of symbol #127. We see that the noise power can cause distortions to the frequency leakage detection. The power of frequency leakage is buried within noise in the case of down-sampling (see Fig.8(b)), from which we may not be able to reliably extract the fingerprint and determine the timing of frequency leakage. Therefore, a natural question arises: *how can we reliably extract the fingerprint of symbol under lower sampling-rates?* We answer the question in the following.

D. Phase-Based Ambiguity Mitigation

In this subsection, we explore the opportunity to enhance the robustness of fingerprint extraction under lower sampling-rates. We propose a phase-based approach and demonstrate the advantages in comparison with the frequency-based approach.



(a) Raw phase data of symbol #127 (b) Phase diff. of adjacent samples

(c) Phase diff. of undersampled chirp (d) Extracted phase jitters

Fig. 9. (a) Raw phase data of a chirp; (b,c) The difference of phases between adjacent samples in fully-sampled and under-sampled chirps; (d) Phase jitters extracted from under-sampled chirps: ambiguity symbols experience phase shifts at different timing.

Opportunity. We explore opportunity for fingerprint extraction by investigating the phase characteristic of LoRa chirps. Recall that LoRa modem employs angle modulation, as shown in Fig.6, the modem controls the phase of each sample to modulate a chirp signal. Denote the phase of the k'th sample by ϕ_k. The value of ϕ_k is determined by the phase of the former sample (*i.e.*, ϕ_{k-1}) and the instantaneous frequency to be modulated (denoted by ω_k). Specifically, $\phi_k = \phi_{k-1} + \omega_k$. Taking phase noise into accounts, we have

$$\phi_k - \phi_{k-1} = \omega_k + J(k), \quad k = 0, 1, \ldots, 2^{SF} - 1 \tag{7}$$

where $J(k)$ denotes the phase jitters added by LoRa modem.

Fig.9(a) presents the raw phase data of symbol #127 when sampling at full clock-rate. According to Eq. (7), the phase difference of adjacent samples represents the instantaneous angular frequency (*i.e.*, ω_n) of chirp modulation (see similarities between Fig.9(b) and Fig.4(b)). The phase difference (*i.e.*, instantaneous frequency) would also suffer from aliasing in the case of under-sampling, as displayed in Fig.9(c).

Based on Eq.(7), we can get the phase of any sample in the chirp as follows.

$$\phi_k = \phi_0 + \sum_{n=0}^{k} \omega_n + \sum_{n=0}^{k} J(n). \tag{8}$$

Eq.(8) indicates that the phase jitters of former samples will be accumulated in the phases of later samples. Suppose that the hardware of LoRa modem only causes phase jitters at a specific time (*i.e.*, denoted by t_{jit}), all samples after t_{jit} then will carry phase jitters yet the samples before t_{jit} will not. As a result, we can anticipate a phase shift of $\Delta\phi_{jit} = \sum_n J(n)$ around the time of t_{jit} (see Fig.9(d)). It enables us to extract fingerprint (*i.e.*, timing of phase jitters) by comparing the phases of all samples in a received chirp, rather than detecting the samples of frequency leakage in short time duration.

Solution. To formally define the problem, we amend the signal model in Eq. (1) to incorporate the phase jitters of LoRa modem. We denote the modulated chirp of symbol f_{sym} as

$$S_{jit}(t, f_{sym}) = S(t, f_{sym}) \cdot e^{j\varphi(t)},\ t \in [0, T_{chirp}) \qquad (9)$$

where $\varphi(t)$ denotes the sum of phase jitters accumulated from time 0 to t (*i.e.*, $\varphi(t) = \sum_{n=0}^{t} J(n)$). Specifically, $\varphi(t) = 0$ if $t < t_{jit}$, otherwise $\varphi(t) = \Delta\phi_{jit}$. The received signal (*i.e.*, $R(t, f_{sym})$) can be modified correspondingly to include phase jitters (*i.e.*, denoted by $R_{jit}(t, f_{sym})$). Similar to Eq. (3), $R_{jit}(Dk)$ represents the samples of a chirp signal received at the down-clocking factor of D. Our goal is to detect time t_{jit} from $R_{jit}(Dk)$, using the phase of received samples.

To extract phase jitters (*i.e.*, $e^{j\varphi(t)}$) from a received chirp (*i.e.*, $R_{jit}(t, f_{sym})$), we can ideally multiply the conjugate of $S(t, f_{sym})$ with the chirp signal based on Eq. (9). However, it requires $S(t, f_{sym})$ which is not available since f_{sym} is still unknown. Recall that f_{alias}, *i.e.*, the aliased frequency of f_{sym}, has been derived using the method in Section III-B. We can locally generate the chirp signal of $S(t, f_{alias})$ with Eq. (1). Since $f_{sym} = f_{alias} + n\frac{BW}{D},\ n = 0, \pm 1, \ldots, \pm\frac{D}{2}$, we multiply the conjugate of $S(t, f_{alias})$ with the received chirp signal, which produces the following.

$$\begin{aligned} &R_{jit}(t, f_{sym}) \cdot S^{-1}(t, f_{alias}) \\ &\quad \approx h(t)e^{j2\pi\Delta f_{cfo}t} S(t, f_{sym}) e^{j\varphi(t)} \cdot S^{-1}(t, f_{alias}) \\ &\quad = h(t)e^{j2\pi\Delta f_{cfo}t} \cdot e^{j2\pi(n\frac{BW}{D})t} \cdot e^{j\varphi(t)} \end{aligned} \qquad (10)$$

In practice, we select downclocking factor D to ensure $n\frac{BW}{D}$ is an integer. Consequently, Eq. (10) is composed of only phase jitters (*i.e.*, $e^{j\varphi(t)}$) and the item of carrier frequency offsets (*i.e.*, $e^{j2\pi\Delta f_{cfo}t}$). Fig.9(d) shows the results of applying Eq. (8) to the under-sampled chirps of two ambiguity symbols #127 and #0. We observe that phase shifts (which indicates the phase jitters of LoRa modem) happen at different time for the two symbols. We can detect the continuity of obtained phase results to extract the timing of phase jitters (*i.e.*, t_{jit}), which forms the fingerprint of symbol encoded on the chirp.

Remarks. The phase-based approach has two advantages over the frequency-based approach: (1) The phase-based approach makes use of all samples in a received chirp, rather than only the samples of short frequency leakage. It improves fingerprint extraction from 'point' detection to 'line' detection, which is more robust in the case of under-sampling. (2) The frequency-based approach performs FFT at different offsets to detect frequency leakage. The computational overhead is $O(N_s \cdot n \log n)$, where n denotes the size of FFT window and N_s the number of samples in a received chirp. In contrast, the phase-based approach is more lightweight (*i.e.*, $O(N_s)$) as it performs no FFTs.

E. Decoding Below the Nyquist

A under-sampled chirp can get decoded as follows. First, we employ Eq. (4) to demodulate the under-sampled chirp, which produces an aliased frequency (*i.e.*, f_{alias}). We next check the phase of received samples to extract the embedded fingerprint of symbol (*i.e.*, t_{jit}, timing of phase jitters). We exploit t_{jit} to estimate the chunk ID (*i.e.*, n_c) of symbol f_{sym} (see the aliasing model in Fig.3). Lastly, by combining f_{alias} and n_c, we recover the encoded symbol as $f_{sym} = n_c\frac{BW}{D} + f_{alias}$.

Algorithm 1 Downclocked LoRa Reception

```
Input: $SF$, $BW$, and downclocking factor $D$.
Output: The payload data of a LoRa packet.
Detect and receive packet at the sampling-rate of $\frac{BW}{D}$.
Synchronize the symbol timing of LoRa packet by searching for chirp boundary in the preamble.
for the chirp of each symbol in the payloads do
    Demodulate $f_{alias}$ from the chirp signal using Eq. (4).
    Detect $t_{jit}$ from the phase of received signal.
    Determine the chunk # of $f_{sym}$ as $n_c = \lfloor\frac{\frac{BW}{2} - \alpha \cdot t_{jit}}{\frac{BW}{D}}\rfloor$.
    Demodulate the symbol as $f_{sym} = f_{alias} + n_c\frac{BW}{D}$.
end for
Decode payload data from $\{f_{sym}\}$.
```

We iteratively apply the above operations to every chirp of a received packet. The demodulated symbols will be fed into a conventional LoRa decoder, which interprets the data transmitted in the packet. We present the detailed scheme of downclocked LoRa reception in Algorithm 1.

Overhead analysis. The computation overheads of symbol demodulation mainly come from FFTs. Let N_s denote the number of samples in a normally received LoRa symbol (*i.e.*, at full clock-rate). The overhead of demodulating a fully-sampled symbol would be $O(N_s \log(N_s))$. In contrast, when the radio down-clocks by a factor of D, the number of received samples of a symbol decreases to $\frac{N_s}{D}$. Performing FFT on the under-sampled chirps requires less computation and the overhead is $O(\frac{N_s}{D}\log(\frac{N_s}{D}))$. Although we need additional computations for fingerprint extraction, it can be completed within $O(\frac{N_s}{D})$ time. The overall computation complexity of Algorithm 1 is $O(\frac{N_s}{D}\log(\frac{N_s}{D}))$, which is *more than D times lower than that of normal demodulation*. Therefore, the algorithm is lightweight and suitable for running on a downclocked MCU that has weaker computation capability.

IV. Downclocking LoRa Receiver

This section studies the impacts of downclocking on LoRa reception pipeline. We present new strategies for packet detection and synchronization, then followed by issues about integration of downclocking to the existing LoRaWAN protocol.

A. Reception Pipeline

The designers of LoRa obviously do not anticipate it would be operated by downclocked radio, so existing approaches on packet reception (such as preamble detecting, timing synchronization, frequency compensation, *etc.*) may not function when downclocked. Before we present the challenges, we first give a brief overview of the reception pipeline of a LoRa receiver.

Packet detection: A receiver will continuously sample the channel to detect any incoming packet. To facilitate packet detection, a LoRa packet is prepended by a preamble consisting of M successive base chirps (*i.e.*, up-chirp with initial frequency $f_0 = -\frac{BW}{2}$, $M = 8$ by default). After the preamble, a Start Frame Delimiter (SFD) composed of 2.25 down-chirps is followed to indicate the start of packet payload [4], [15]. Upon detecting a LoRa preamble, the receiver would prepare to receive the following payload chirps of the packet.

Synchronization: To ensure correct decoding of payload, a receiver relies on LoRa preamble to perform timing and frequency synchronization. The M base chirps in the preamble help receiver lock on symbol timing in the payload. Once the boundaries of each symbol are identified, a receiver can use a standard algorithm to demodulate. However, as the oscillator frequency is hardly the same at the sender and receiver, it leaves carrier frequency offset (*i.e.*, CFO) on received signals. A receiver performs CFO estimation on the preamble. It uses the prior knowledge of base chirps to estimate CFO and compensate the frequency of received signals before demodulation (*i.e.*, frequency synchronization).

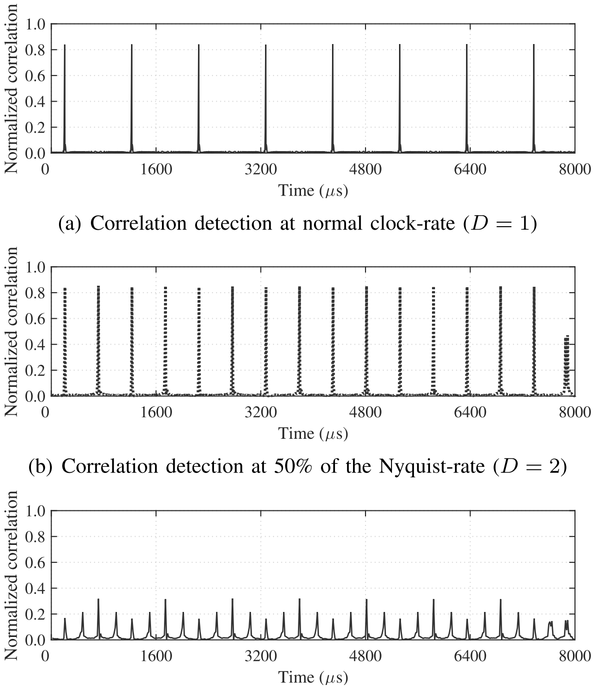
B. Packet Detection

The purpose of packet detection is to detect the presence of LoRa preamble and prepare radio for packet reception. We conventionally identify a preamble by correlating the received signal to a locally-generated base chirp. For a downclocked receiver, we perform correlation on the received under-sampled signal (*i.e.*, $R(Dk)$ in Eq. (3)) as below.

$$corr(Dk) = \frac{\sum_{j=0}^{\lceil \frac{N_s}{D} \rceil - 1} R(D(k+j)) \cdot C^{-1}(Dj)}{\sum_{j=0}^{\lceil \frac{N_s}{D} \rceil - 1} |C^{-1}(Dj)|^2} \qquad (11)$$

where, $C^{-1}(\cdot)$ denotes the conjugate of base chirp, $N_s = 2^{SF}$ is the number of samples of a standard base chirp, and D is the downclocking factor. In particular, Eq. (11) normalizes the result to allow fair comparison of correlation output across the received samples of different downclocking factors.

Fig.10 compares the correlation responses of the same LoRa preamble when received with different sampling rates. We present the result of normal clock-rate in Fig.10(a) as a benchmark, where correlation peaks indicate the boundaries of base chirps. Comparing Fig.10(a) and (b), we see that as sampling rate reduces to 50% of the Nyquist-rate, additional correlation peaks appear in the middle of the peaks of Fig.10(a). The new peaks do not correspond to chirp boundaries in preambles due to the frequency aliasing effect. For example, with the sampling rate of 50% Nyquist-rate, the chirps of symbol #0 and symbol #127 exhibit high resemblance after frequency aliasing, as shown in Fig.4(c,d). If we correlate the under-sampled signal with a base chirp using Eq. (11), we get two peaks—one for symbol #0 (*i.e.*, at real chirp boundaries), and the other for symbol #127 (*i.e.*, in the middle of real boundaries). As we further decrease sampling rate, more peaks appear yet the relationship of correlation becomes rather weak, as shown in Fig.10(c). It is hard to discern correlation peaks from noise.



(a) Correlation detection at normal clock-rate ($D = 1$)

(b) Correlation detection at 50% of the Nyquist-rate ($D = 2$)

(c) Correlation detection at 25% of the Nyquist-rate ($D = 4$)

Fig. 10. Correlation responses on the raw PHY samples of LoRa preamble with normal and sub-Nyquist sampling rate: *fail to extract symbol timing from sub-sampled preamble using correlation detection.*

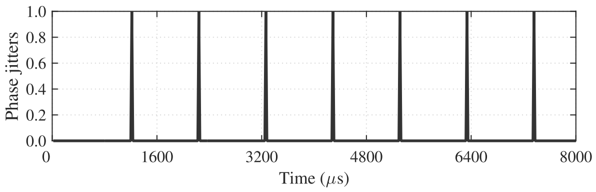


Fig. 11. The timing of phase jitters detected on the raw samples of LoRa preamble received with 25% of the Nyquist-rate ($D = 4$), we obtain the same results with other down-sampling rates. Note that the locations of phase jitters are slight different from the correlation results in Fig.10(a) because two approaches operate differently: phase jitters are detected at the end of a base chirp, yet the correlation peaks are detected at the start of a base chirp.

In the following, we propose a new approach to reliably detect packets with a downclocked receiver. We leverage the observation that the timing of phase jitters uniquely identifies a LoRa chirp (*i.e.*, symbol fingerprint). As LoRa preamble is composed of repetitive base-chirps, phase jitters are anticipated to appear at around chirp boundaries (see Fig.5). We extract samples of a chirp duration from received signal (*i.e.*, *observation window*) and detect phase jitters of the chirp signal in the window. By moving the window across chirps, we iteratively detect all phase jitters in the received signal. If the incoming signal is a LoRa preamble, we obtain identical fingerprints in successive windows. As shown in Fig.11, the detected phase jitters appear uniformly in time with fixed interval. We obtain the same results across different downclocking factors. In particular, the results are clear (*i.e.*, noise-free) and more reliable as compared with the correlation results in Fig.10(c). In contrast, if the received signal corresponds to packet payload, we will detect different fingerprints over different chirps. The detected phase jitters appear non-uniformly in time. We can detect the repetition pattern of phase jitters to differentiate preamble from the payload of packet.

C. Synchronization

We next use preamble to perform frequency and timing synchronization. Timing synchronization detects symbol timing

Algorithm 2 Symbol Timing Synchronization

```
Input:  Detected timing of phase jitters $t_{jit}$,
        The maximum search range $t^*$.
Output: The timing of base-chirp boundary in preamble.
Upon detecting a preamble:
Estimate $\Delta f_{cfo}$ based on Eq. (10).
for each sample in time range $[t_{jit} - T_{chirp} - t^*, t_{jit} - T_{chirp} + t^*]$ do
    Extract the chirp starting from the current sample.
    Remove $\Delta f_{cfo}$ from the received samples of the chirp.
    Demodulate $f_{alias}$ from the chirp signal using Eq. (12).
    if current $f_{alias}$ is smaller than the former one then
        Record the current sample position as chirp boundary.
    end if
end for
```

from preamble and uses the information to pinpoint symbols in the payload; while frequency synchronization estimates and compensates CFO for the received signal. In particular, as downclocking enlarges the time interval of sampling, the offset between sampled symbol edge and the real edge can translate to corresponding offset in frequency for symbol demodulation, termed Sampling Frequency Offset (SFO). The operation of synchronization should also take SFO into account.

Timing synchronization. We normally use correlation to detect chirp boundaries from preamble (see Fig.10). However, the method fails for under-sampled preamble. In case of downsampling, we can detect phase jitters in received preamble as shown in Fig.11 and infer chirp boundaries from the timing of phase jitters. Here, a key question arises: *does the timing of phase jitters indicate the correct symbol timing?*

Ideally, phase jitters are expected to be right at the boundary of base-chirps. However, we observed that the timing of phase jitters deviates from their ideal locations and the deviations vary across different nodes, due to hardware imperfection. Without loss of generality, let Δt_{hw} denote the deviation of phase jitters from the ideal timing, and t_{jit} be the detected timing of phase jitters. We need to remove Δt_{hw} from t_{jit} to extract the correct symbol timing.

We search over a small range around t_{jit} to look for the correct symbol timing. For each sample in the range, we demodulate the chirp starting from the sample with Eq. (4). The obtained frequency $f_{alias} = \mod(f_0 + \Delta f_{sample} + \Delta f_{cfo}, \frac{BW}{D})$ is minimized if the chirp is extracted from the correct timing. In practice, we can first remove CFO (*i.e.*, Δf_{cfo}) from chirp signal to achieve more precise detection of symbol timing. Δf_{cfo} can be estimated at the time of phase jitter detection (*i.e.*, the slope of phase change in Fig.9(d) corresponds to Δf_{cfo}). We describe the key steps of symbol timing synchronization in Algorithm 2. Noting that the search range (t^*) is usually small. In most of our experiments, we only need to search up to three samples to find the correct symbol timing. The overhead would not be a problem.

Frequency compensation. Now we can precisely pinpoint under-sampled symbols in the payload. We still need to remove CFO and SFO to correctly demodulate a symbol. CFO and SFO would cause demodulation errors if $(\Delta f_{sample} + \Delta f_{cfo})$ is larger than the frequency of one FFT bin (*i.e.*, $\frac{BW}{2^{SF}}$). We observe that Δf_{sample} and Δf_{cfo} remain stable across chirps of the same packet. We can employ a conventional method for CFO and SFO cancellation by calibrating the chirp of payload symbol (*i.e.*, Eq. (4)) with a preamble base-chirp $R_{base}(Dk)$ as below.

$$\frac{R(Dk)\cdot C^{-1}(\frac{Dk}{BW})}{R_{base}(Dk)\cdot C^{-1}(\frac{Dk}{BW})} \approx \frac{e^{j2\pi(f_{sym}+\Delta f_{sample}+\Delta f_{cfo})\cdot\frac{Dk}{BW}}}{e^{j2\pi(f_{base}+\Delta f_{sample}+\Delta f_{cfo})\cdot\frac{Dk}{BW}}} = e^{j2\pi(f_{sym}-f_{base})\cdot\frac{Dk}{BW}} \qquad (12)$$

where, $f_{base} = -\frac{BW}{2}$ is the initial frequency of base chirp. We can write the left side of the equation as $R(Dk) \cdot R_{base}^{-1}(Dk)$, here $R_{base}^{-1}(Dk)$ is the conjugate of the under-sampled signal of base chirp from preamble. With Eq. (12), we remove CFO and SFO from $R(Dk)$. We finally perform FFT on Eq. (12) to demodulate the under-sampled chirp, which produces the aliased frequency of f_{sym} (*i.e.*, f_{alias}).

D. Scheduling of Downclocking

We now present *LiteNap* module that schedules the downclocking of LoRa radio. We specifically focus on (1) *how to determine the downclocking factor of a receiver* and (2) *how to integrate downclocking into the existing protocol.*

Downclocking factor prediction. The selection of downclocking factor D shall trade off between power consumption and reliability of packet reception. A LoRa receiver needs to jointly consider the power budget as well as channel conditions to optimize reception performance. Essentially, LoRa radio requires a minimum Signal-to-Noise Ratio (SNR) to decode packet. The receive sensitivity varies with the configurations of LoRa packet (*e.g.*, SF and BW). According to the datasheet of Semtech [9], SX1276 radio is able to receive signal with power strength as low as -148 dBm, even with negative SNRs (*i.e.*, below the noise floor), which leaves sufficient link margin for downclocking. As such, we coarsely predict downclocking factor as follows.

$$D = 2^{\lfloor \frac{SNR - Sensitivity}{3\text{dB}} \rfloor} \qquad (13)$$

where, SNR and $Sensitivity$ represent the SNR of received packet and the receive sensitivity of radio, respectively. The receiver sensitivity can be obtained from radio's datasheet.

In practice, a receiver radio (*e.g.*, SX1276) will measure the SNR of received packet and record in a local register. We can access the measured SNR and use Eq. (13) to coarsely predict an initial configuration of downclocking factor. We maintain a history record of the measured SNRs as well as the error rates of packet reception at each node. If SNR decreases and packet error rate exceeds a threshold for three successive packets, we will double the clock rate to improve the reliability of packet reception. On the other hand, if SNR improves over 3 dB for three successive packets, we will reduce the clock rate by half for power saving.

Managing downclocked mode. *LiteNap* introduces a novel downclocked work mode for LoRa radio. It integrates the new

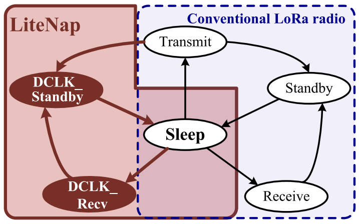


Fig. 12. State transitions of *LiteNap*.

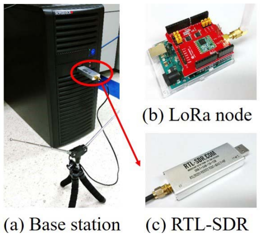

(a) Base station (b) LoRa node (c) RTL-SDR

Fig. 13. Experiment devices.

downclocked mode into existing protocol to comprehensively manage the scheduling of radio activities.

Modern radio generally supports duty-cycling that toggles ON&OFF components for power management depending on radio activities. Radio consumes ultra-low power in Sleep state (*e.g.*, 0.2 μA for SX1276) by shutting down most components when they are not in use, and powers on the components on-demand for Transmit and Receive. For example, as shown in Fig.12, a conventional LoRa radio would transit among various states to realize duty-cycled operations. The LoRaWAN protocol defines three classes, *i.e.*, Class A for stochastic duty-cycled communication, Class B for synchronous communication and Class C for continuous communication, that manage the duty-cycling of radio with different strategies. Different from the existing strategy of ON-OFF power management, *LiteNap* adopts a new approach that operates LoRa radio in a light sleep mode without affecting the basic functionalities.

Specifically, we define two new states (*i.e.*, DCLK_Standby and DCLK_Recv) and integrate into the control flow of LoRa radio as shown in Fig.12. DCLK_Standby and DCLK_Recv correspond to the downclocked operation states of the conventional Standby state and Receive state of LoRa radio. *LiteNap* does not downclock radio in Transmit state. It transmits packet at full clock-rate so as to interact with coexisting LoRa devices that employ no downclocking. Essentially, by extending radio with the two states (*i.e.*, DCLK_Standby and DCLK_Recv), *LiteNap* provides an opportunity to adapt clock rates so as to reduce the operational power consumption of radio in Standby and Receive.

We can integrate *LiteNap* into the LoRaWAN protocol. Let us take Class A as an example. As shown in Fig.12, *LiteNap* shares Sleep and Transmit states with a conventional scheduler. If the node wants to send a packet, the LoRaWAN protocol schedules a Class A TX window, which wakes up radio from sleep and transits to Transmit state. *LiteNap* takes over the schedule after packet transmission. It will transit radio to DCLK_Standby in the rest of TX window. The radio returns to Sleep when TX window is over. LoRaWAN will schedule two RX windows after a certain delay. *LiteNap* takes over the control by transiting radio from Sleep to DCLK_Recv state and operating radio to receive at lower clock-rate. The radio will transit to DCLK_Standby after packet reception.

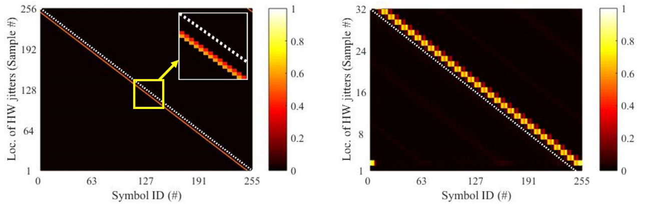
E. Discussion

Targeted scenario and the gains. *LiteNap* aims to improve the energy efficiency of LoRa nodes rather than gateways. We reduce the power consumption of node radio in both Receive and Standby states by decreasing the clock-rate. The clock-rate of radio in Transmit state is not reduced. Although IoT nodes generally transmit more receive less, they can benefit from *LiteNap* because the LoRaWAN protocol schedules mandatory RX windows for downlink traffics after uplink transmission. *LiteNap* reduces the operational power of LoRa node in RX windows. In particular, it enables the node to reliably detect and receive packet at the same downclocked rate without any switch overhead. We can expect more power gains for LoRa nodes with downlink-dominant traffics, where nodes running with LoRaWAN Class B or C spend more time in the Standby and Receive states than the nodes of Class A.

Transmission optimizing. We cannot transmit LoRa chirps while downclocked. As a matter of fact, the dominant power source of radio on the TX chain is PA (*i.e.*, power amplifier) rather than DAC (*i.e.*, Digital-to-Analog Converter). Reducing clock-rate does not reduce much power for transmission. The energy consumption of transmission is mainly determined by the transmit power and air time of packet. An effective way is to choose the best configurations of LoRa communication (*e.g.*, frequency [16], spreading factor and bandwidth [17]) that can reduce the transmit power and air time of packet for energy optimizing. One can also employ multi-antenna and beamforming techniques at a gateway for SNR enhancement (*e.g.*, Charm [18]) which allows LoRa nodes to transmit with more power-efficient parameters.

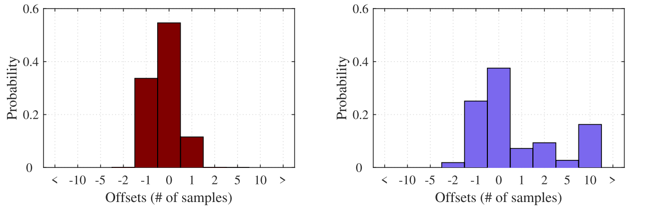
V. Evaluation

We implement *LiteNap* on GNURadio based on `gr-lora` projects [19] and build a testbed using COTS LoRa nodes and software radio base station to evaluate the proposed schemes, as displayed in Fig.13. The LoRa nodes are composed of HopeRF's RFM96W transceiver with Semtech SX1276 radio chip. As the clock rate of SX1276 is fixed at 32 MHz, we use a low-cost software defined radio (*i.e.*, RTL-SDR dongle) to receive the packets transmitted by COTS LoRa nodes. The low-cost SDR is used to only receive but not transmit LoRa packets. We re-sample the received signal to emulate the downclocked reception at specific clock-rates. We implement our LoRa decoding scheme based on the GNURadio library and develop MATLAB program to process the PHY samples. We operate SDRs and LoRa nodes at the 915MHz frequency. The measured noise level is around -90dBm throughout the experiments. Unless otherwise specified, we configure LoRa nodes with $SF = 8$, $BW = 250$ kHz and coding rate $CR = 4/5$. Note that the power saving strategy of LiteNap (*i.e.*, down-clocking) is complementary to the



(a) (b)

Fig. 14. The fingerprints of all 256 symbols extracted from (a) fully-sampled chirps and (b) under-sampled chirps (downclocking factor $D = 8$), respectively. The color indicates the probability of phase jitters detected at different sample locations within the chirps of different symbols.



(a) Phase-based approach (b) Frequency-based approach

Fig. 15. Histogram of offsets (in # of PHY samples) between the extracted fingerprints and the ideal locations under the downclocking factor of $D = 8$.

existing duty-cycling strategy employed by LoRaWAN. We set the duty-cycle of LoRa nodes to 2% for the purpose of performance evaluation.

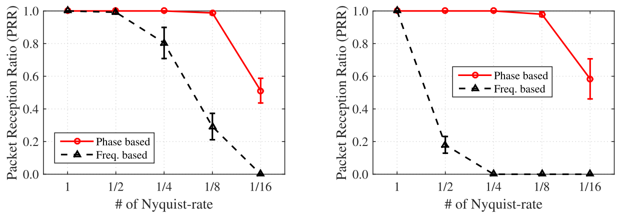
A. Fingerprint Extraction

In this experiment, we study the characteristics of fingerprint and evaluate the performance of the proposed approaches for fingerprint extraction. We use a USRP to receive the packets transmitted by COTS LoRa nodes. The packet payload consists of 37 PHY symbols. We randomly choose the payload contents and collect more than 10,000 packets to obtain sample data for all 256 symbols.

We use the phase-based approach to extract fingerprints (*i.e.*, locations of hardware phase jitters) from received chirps and associate them with corresponding symbol IDs. Fig.14 shows the distribution of the jitter locations of all 256 symbols. As expected, the jitter locations change linearly with symbol IDs. It verifies that the location of phase jitters can be used as fingerprints to uniquely identify the symbols.

Fig.14 also compares the extracted fingerprints of symbols with their expected locations as marked by the white dashed line. We observe a tiny offset between the obtained results and ideal locations. This offset varies with different LoRa nodes. But it remains constant across symbols of the same device. In practice, we can detect the offset from the preamble of LoRa packets and subtract it from the extracted jitter locations of payload symbols to obtain the correct fingerprints.

Fig.15 presents the histogram of offsets between the extracted fingerprints after being corrected with the preambles and the ideal locations. We compare the phase-based and frequency-based approaches across ten LoRa nodes with the downclocking factor of 8. As the frequency-based approach is vulnerable to noise, lots of detected fingerprints deviate away from ideal locations. In contrast, the phase-based approach is more robust. All detected fingerprints are within one sample of the ideal locations and can be used to resolve ambiguity, since aliased symbols are separated by more than one sample.



(a) Good SNRs (b) Poor SNRs

Fig. 16. Impacts of downclocking on Packet Reception Ratio (PRR).

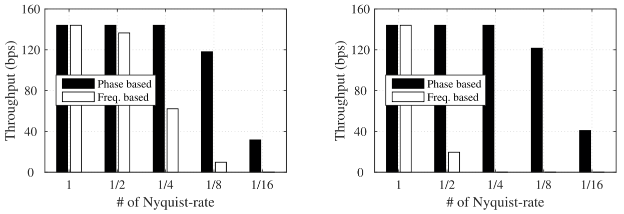
B. Packet Reception Performance

1) Basic Performance: This experiment examines the communication performance of *LiteNap*. We setup a COTS LoRa node as the transmitter and a RTL-SDR dongle as the receiver. We configure LoRa node to transmits 1,000 packets (payload: 22 Bytes). RTL dongle receives at the physical sampling-rate of 1 Msps. We re-sample the received signal to emulate downclocked reception. We vary the re-sampling rate across 1, 1/2, 1/4, 1/8 and 1/16 of the Nyquist-rate. In particular, we repeat the experiments across five LoRa nodes and two RTL-SDR dongles. We change the locations of transmitter within a 8-floor building, where the maximum transmission range can be about 150m in Non-Line-Of-Sight (NLOS), and classify received packets into two SNR regimes, *i.e.*, Poor (<5dB) and Good (≥5dB) SNRs.

We first evaluate the Packet Reception Ratio (PRR), *i.e.*, ratio of packets being successfully decoded. Fig.16 presents the PRR of *LiteNap* in different SNRs. We see that the frequency based approach produces lower PRRs. The performance of frequency based approach degrades fast when receiver downclocks to lower rates. As the SNRs become poor, the frequency-based approach cannot decode any packet when clock-rate falls below 1/4 the Nyquist's, as shown in Fig.16(b). In comparison, the phase-based approach outperforms the frequency-based approach in both good and poor SNRs. The phase-based approach can correctly decode all packets as the receiver downclocks to 1/4 the Nyquist-rate. The PRR remains above 95% when the clock-rate is 1/8 the Nyquist-rate.

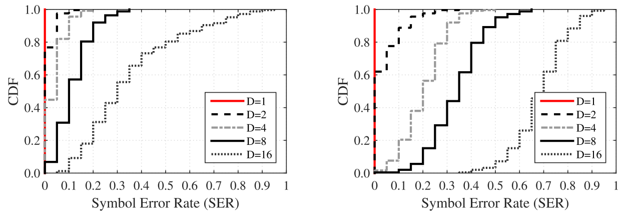
The throughput of *LiteNap* exhibits a similar trend with that of PRR. As shown in Fig.17, the throughput of the phase-based approach does not change as the receiver downclocks from 1 to 1/4 the Nyquist-rate. It still achieves about 85% the throughput of full clock-rate when the clock-rate is 1/8 the Nyquist-rate. The throughput of the frequency-based approach decreases fast as clock-rate reduces to sub-Nyquist rates or when the SNRs become poor (*e.g.*, <5dB).

Fig.18 examines the link-layer symbol demodulation errors of *LiteNap* in good SNRs (we omit the results of poor SNRs due to page limit). We see that the Symbol Error Rates (SERs) of both approaches become higher as the downclocking factor (*i.e.*, D) increases. It is because demodulation errors stem



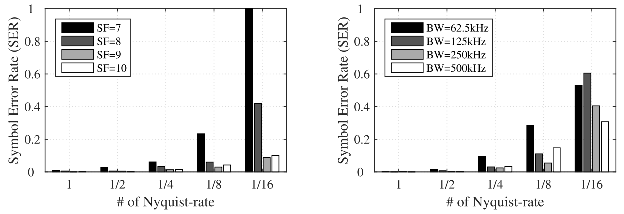
(a) Good SNRs (b) Poor SNRs

Fig. 17. Impacts of downclocking on throughput.



(a) Phase-based approach (b) Frequency-based approach

Fig. 18. Performance of downclocked symbol demodulation in good SNRs.



(a) SER vs. SF (BW=250kHz) (b) SER vs. BW (SF=8)

Fig. 19. Impacts of packet configuration on downclocked symbol demodulation (phase-based approach) with good SNRs.

mainly from the detection error of symbol fingerprint, which can cause *LiteNap* to incorrectly demodulate aliased symbols. As the downclocking factor increases, the number of ambiguity symbols would increase, which in turn increases the SERs of *LiteNap*. Nevertheless, *LiteNap* can use the phase-based approach to recover 99% symbols with less than 0.1 Symbol Error Rates (SERs) when downclocking factor $D = 2$. The ratio of SER ≤ 0.1 decreases to 80% and 60% when $D = 4$ and 8. Despite that, most demodulation errors can be corrected by the error correction scheme of LoRa (*i.e.*, Hamming coding [20]). This explains why the PRRs of phase-based approach are constantly above 95% when the receiver downclocks to 1/2, 1/4 and 1/8 of the Nyquist-rate (see Fig.16(a)).

2) *Impact of LoRa Packet Configuration:* The configurations of LoRa packet affect reception performance. We next investigate the impact of LoRa parameters such as spreading factor (SF) and bandwidth (BW) on downclocked packet reception. Without otherwise stated, we adopt the same settings as the above experiments. We employ the phase-based approach to demodulate symbols; and we compare the symbol error rates (SERs) of different SF and BW configurations under various downclocking factors. We only present the evaluation results of good SNRs in Fig.19. The results of poor-SNR exhibit a similar trend (not presented).

In the first experiment, we set LoRa bandwidth to 250kHz and change SF from 7 to 10. Fig.19(a) displays the relationship between SER and SF. We see that packets with larger SFs are more robust to downclocking. To be specific, a larger SF generally produces lower SER under the same downclocking factor (D). For example, when we receive packet with 1/8 of the Nyquist-rate, the average SER is 0.23 for SF = 7, while the number decreases to 0.04 as SF increases to 10. As we further reduce clock-rate to 1/16 Nyquist-rate, we cannot correctly recover symbols from the undersampled packet of $SF = 7$, whereas the packets of SF = 9 and SF = 10 can still get decoded with less than 0.1 symbol error rates. It is because larger SF corresponds to longer symbol duration, which is more resistant to the SNR losses caused by downclocking.

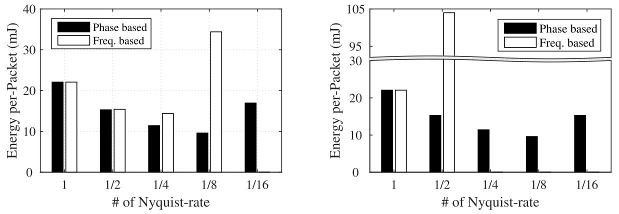
Fig.19(b) evaluates the implication of LoRa bandwidth. We vary BW from 62.5kHz to 500kHz while setting SF = 8. As we can see, a larger bandwidth is more robust to downclocking. For instance, when clock-rate is 1/8 of the Nyquist-rate, the average SER is 0.29 for BW = 62.5kHz yet it decreases to 0.05 for BW = 250kHz. The reason is as the bandwidth becomes wider, the frequency gap between LoRa symbols (*i.e.*, $\Delta f = \frac{BW}{2^{SF}}$) increases. A larger gap helps the differentiation of symbols in the frequency domain, which can reduce errors of demodulating f_{sym} as another symbol (*e.g.*, $f_{sym} + \Delta f$).

C. Energy Saving

We evaluate the energy efficiency of downclocked reception through trace-driven simulations. We characterize the power consumption of LoRa radio based on the model proposed in [4]. The basic energy profiles are obtained from the data sheet of Semtech SX1276 [9]. We use the power of SX1276 as an estimate for the power consumption of fully-clocked radio. The downclocked power consumption can be estimated by proportionally scaling the power of full-clocking with respect to the downclocking factor (*i.e.*, D). We summarize the adopted configurations in Table II.

We collect traffic traces from a realistic LoRaWAN link composed of one transmitter (*i.e.*, COTS LoRa node) and one receiver (*i.e.*, RTL-SDR base station). Both transmitter and receiver run the LoRaWAN MAC protocol with Class A. We configure the LoRa node to periodically send packets (payload: 22 Bytes) with a duty-cycle of 2%. The RTL-SDR dongle employs *LiteNap* to decode received packets. If a packet is successfully decoded, the receiver will reply with an ACK. A packet would be re-transmitted if no ACK is received. We vary downclocking factor D across 1, 2, 4, 8, 16, and receive 1,000 packets under each downclocking setting. We replay the collected traffic to simulate the energy drains of both transmitter and receiver. We compute *the total energy consumption of transmitter and receiver*, then average it out to the number of received packets. We use this per-packet energy consumption as a metric to evaluate the energy-efficiency of downclocked LoRa reception.

Fig.20 presents the energy performance of *LiteNap*. The phase-based approach exhibits similar performance regardless of the SNR conditions. The best performance is achieved when $D = 8$. In the case of good SNRs (see Fig.20(a)), as receiver downclocks from the Nyquist-rate to 1/8 Nyquist, the per-packet energy consumption decreases from 22.1 to 9.6 mJ (*i.e.*, reduced by 56.6%). However,



(a) Good SNRs

(b) Poor SNRs

Fig. 20. Per-packet energy consumption under different downclocking factors. For the frequency-based approach, since no packets are correctly received in the cases of $D = 16$ in good SNRs and $D = 4$, 8, 16 in poor SNRs (see Fig.16), the corresponding results are absent.

TABLE II

POWER CHARACTERISTICS OF DOWNCLOCKED LORA RECEPTION

Downclocking factors	***D*=1**	***D*=2**	***D*=4**	***D*=8**	***D*=16**
Transmit power (mW)	66.00	66.00	66.00	66.00	66.00
Receive power (mW)	37.95	24.67	17.08	13.28	11.39
Standby power (mW)	5.94	3.86	2.67	2.08	1.78
Packet on-air time	35.84 ∼ 46.08 (ms)				
LoRaWAN on-duty time*	3 (s)				

*on-duty time = TX Win + 2×RX Win + RX Delays(idle waiting).

the energy increases when the clock-rate decreases to 1/16 the Nyquist-rate, due to an increasing number of packet re-transmissions. On the other hand, the frequency based approach produces the optimal performance when $D = 4$ in good SNRs, with an energy reduction of 36.5% as compared to that of $D = 1$. However, the frequency based approach cannot reduce power consumption in poor SNRs due to re-transmissions caused by packet errors, while the phase based approach can still reduce power in poor SNRs.

VI. RELATED WORK

Energy efficiency for LoRa. Many prior efforts [4], [21]–[24] had been devoted to characterize LoRa power consumption. They empirically measure the power consumption of COTS LoRa radio in various operation modes and LoRaWAN classes. Based on the measurements, researchers propose energy models for LoRa and analytically study the relationships between energy consumption and various impacting factors, such as network topology [21], duty-cycle ratio [4], configuration of communication parameters [24], *etc*. These works conclude that the energy performance of current LoRa platforms are far from optimal [4], [22], [25].

Existing works study resource scheduling [17], [26]–[28] and parameter allocations to reduce power consumption [29]. Liando *et al.* [4] employ prediction models to allocate spreading factor and transmission power for LoRa. Bor and Roedig [30] propose a link probing scheme to select the optimal LoRa parameters. uLoRa [31] presents an ultra low-power hardware and software design. References [32]–[35] employ backscatter communications to reduce the power consumption of LoRa hardware to μW level. Orthogonal to these works, this paper aims to reduce the power consumption by downclocking LoRa radio.

Downclocked communication. E-MiLi [8] downclocks WiFi radio during idle listening to reduce power consumption. SASD [36] extends E-MiLi by conveying data on preamble that can be decoded by a down-clock radio. SloMo [37] applies compressed sensing to DSSS to enable downclocked sending and receiving. Enfold [38] exploits the aliasing structure of OFDM to decode under-sampled packets. Sampless WiFi [39] combats the aliasing of OFDM by combining multiple re-transmissions to recover packet data. Recently, LongBee [40] employs downclocking to concentrate signal power at narrower bandwidth for long-range communications across WiFi and ZigBee. PLoRa [34] detects LoRa packets at sub-Nyquist sampling-rates, but it does not support the decoding of under-sampled packets. To the best of our knowledge, this paper is the first study on downclocked LoRa packet reception.

VII. CONCLUSION

This paper aims to improve the energy efficiency of LoRa by enabling sub-Nyquist sampling and packet decoding. To this end, we study the frequency aliasing of under-sampled LoRa packets, which reveals that the frequency aliasing can cause ambiguity in demodulation. Fortunately, our empirical study also discovers that the timing of frequency leakage within a chirp can serve as a fingerprint to uniquely identify a symbol. More importantly, the timing information can be well reserved even when under-sampled below the Nyquist sampling rate. Based on this observation, we propose two fingerprint extraction methods to reliably detect the timing information and resolve the ambiguity caused by frequency aliasing. We update the reception pipeline of LoRa radio to enable reliable packet detection and decoding when operated in downclocked mode. We evaluate the proposed methods through testbed experiments and trace-driven simulations. Results show that a down-clock receiver can reduce power consumption by up to 50%, while achieving comparable packet reception performance of a full-clock receiver in good channel conditions.

REFERENCES

[1] SigFox. (Jan. 2019). *SigFox Overview*. [Online]. Available: https://www.sigfox.com/en/sigfox-iot-technology-overview

[2] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "NB-IoT system for M2M communication," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, New York, NY, USA, Apr. 2016, pp. 1–5.

[3] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 371–388, 1st Quart., 2020.

[4] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa: Experiences from a large-scale measurement study," *ACM Trans. Sen. Netw.*, vol. 15, no. 2, pp. 16:1–16:35, Feb. 2019, doi: 10.1145/3293534.

[5] Semtech. (Jun. 2019). *Semtech SX1276: 137MHz to 1020MHz Long Range Low Power Transceiver*. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276

[6] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," in *Proc. Int. Conf. Embedded Wireless Syst. Netw.*, Kimble County, TX, USA, 2016, pp. 361–366. [Online]. Available: https://dl.acm.org/citation.cfm?id=2893711.2893802

[7] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "Evaluating the LoRaWAN protocol using a permanent outdoor testbed," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4726–4733, Jun. 2019.

[8] X. Zhang and K. G. Shin, "E-MiLi: Energy-minimizing idle listening in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1441–1454, Sep. 2012.

[9] Semtech. (Jun. 2019). *SX1276/77/78/79 Datasheet*. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276

[10] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, “A case for adapting channel width in wireless networks,” in *Proc. ACM SIGCOMM Conf. Data Commun. (SIGCOMM)*, New York, NY, USA, 2008, pp. 135–146, doi: 10.1145/1402958.1402975.

[11] A. Demir, A. Mehrotra, and J. Roychowdhury, “Phase noise in oscillators: A unifying theory and numerical methods for characterization,” *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 47, no. 5, pp. 655–674, May 2000.

[12] F. A. P. Figueiredo, F. S. Mathilde, F. L. Figueiredo, and F. A. C. M. Cardoso, “An FPGA-based time-domain frequency shifter with application to LTE and LTE—A systems,” in *Proc. IEEE 6th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb. 2015, pp. 1–4.

[13] S. Kim and J.-W. Chong, “Chirp spread spectrum transceiver design and implementation for real time locating system,” *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 8, Aug. 2015, Art. no. 572861, doi: 10.1155/2015/572861.

[14] Semtech. (Jul. 2019). *SX1272/3/6/7/8: LoRa Modem Designer’s Guide AN1200.13*. [Online]. Available: https://www.semtech.com/uploads/documents

[15] X. Xia, Y. Zheng, and T. Gu, “FTrack: Parallel decoding for LoRa transmissions,” in *Proc. 17th Conf. Embedded Netw. Sensor Syst.*, Nov. 2019, pp. 192–204.

[16] A. Gadre, R. Narayanan, A. Luong, A. Rowe, B. Iannucci, and S. Kumar, “Frequency configuration for low-power wide-area networks in a heartbeat,” in *Proc. USENIX NSDI*, 2020, pp. 339–352.

[17] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, “Towards energy-fairness in LoRa networks,” in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 788–798.

[18] A. Dongare *et al.*, “Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks,” in *Proc. 17th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2018, pp. 60–71.

[19] Gr-LoRa GitHub Community. (Jul. 2019). GR-LoRa projects. [Online]. Available: https://github.com/rpp0/gr-lora

[20] P. Robyns, P. Quax, W. Lamotte, and W. Thenaers, “A multi-channel software decoder for the LoRa modulation scheme,” in *Proc. 3rd Int. Conf. Internet Things, Big Data Secur.*, Mar. 2018, pp. 1–11.

[21] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, “Evaluating LoRa energy efficiency for adaptive networks: From star to mesh topologies,” in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8.

[22] L. Casals, B. Mir, R. Vidal, and C. Gomez, “Modeling the energy performance of LoRaWAN,” *Sensors*, vol. 17, no. 10, p. 2364, Oct. 2017, doi: 10.3390/s17102364.

[23] P. S. Cheong, J. Bergs, C. Hawinkel, and J. Famaey, “Comparison of LoRaWAN classes and their power consumption,” in *Proc. IEEE Symp. Commun. Veh. Technol. (SCVT)*, Nov. 2017, pp. 1–6.

[24] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, “Energy consumption model for sensor nodes based on LoRa and LoRaWAN,” *Sensors*, vol. 18, no. 7, p. 2104, Jun. 2018, doi: 10.3390/s18072104.

[25] M. Costa, T. Farrell, and L. Doyle, “On energy efficiency and lifetime in low power wide area network for the Internet of Things,” in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2017, pp. 258–263.

[26] Z. Qin and J. A. McCann, “Resource efficiency in low-power wide-area networks for IoT applications,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.

[27] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Low overhead scheduling of LoRa transmissions for improved scalability,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3097–3109, Apr. 2019.

[28] W. Du, J. C. Liando, H. Zhang, and M. Li, “When pipelines meet fountain: Fast data dissemination in wireless sensor networks,” in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 365–378.

[29] B. Reynders, W. Meert, and S. Pollin, “Power and spreading factor control in low power wide area networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[30] M. Bor and U. Roedig, “LoRa transmission parameter selection,” in *Proc. 13th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Jun. 2017, pp. 27–34.

[31] H. Sallouha, B. Van den Bergh, Q. Wang, and S. Pollin, “ULoRa: Ultra low-power, low-cost and open platform for the LoRa networks,” in *Proc. 4th ACM Workshop Hot Topics Wireless (HotWireless)*, New York, NY, USA, 2017, pp. 43–47, doi: 10.1145/3127882.3127890.

[32] V. Talla, M. Hessar, B. Kellogg, A. Naja, J. R. Smith, and S. Gollakota, “LoRa backscatter: Enabling the vision of ubiquitous connectivity,” in *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 105:1–105:24, Sep. 2017, doi: 10.1145/3130970.

[33] A. Varshney, C. P. Penichet, C. Rohner, and T. Voigt, “Towards wide-area backscatter networks,” in *Proc. 4th ACM Workshop Hot Topics Wireless (HotWireless)*, New York, NY, USA, 2017, pp. 49–53, doi: 10.1145/3127882.3127888.

[34] Y. Peng *et al.*, “PLoRa: A passive long-range data network from ambient LoRa transmissions,” in *Proc. Conf. ACM Special Interest Group Data Commun.*, New York, NY, USA, Aug. 2018, pp. 147–160, doi: 10.1145/3230543.3230567.

[35] M. Hessar, A. Najafi, and S. Gollakota, “NetScatter: Enabling large-scale backscatter networks,” in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*. Boston, MA, USA: USENIX Association, 2019, pp. 271–284. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/hessar

[36] T. Xiong, J. Yao, J. Zhang, and W. Lou, “It can drain out your energy: An energy-saving mechanism against packet overhearing in high traffic wireless LANs,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1911–1925, Jul. 2017.

[37] F. Lu, G. M. Voelker, and A. C. Snoeren, “SloMo: Downclocking WiFi communication,” in *Proc. 10th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2013, pp. 255–258.

[38] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren, “Enfold: Downclocking OFDM in WiFi,” in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 129–140.

[39] W. Wang, Y. Chen, and L. Wang, “Sampleless Wi-Fi: Bringing low power to Wi-Fi communications,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1663–1672, Jun. 2017.

[40] Z. Li and T. He, “LongBee: Enabling long-range cross-technology communication,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 162–170.

Xianjin Xia (Member, IEEE) received the B.S., M.Sc., and Ph.D. degrees in computer science from Northwestern Polytechnical University, Xi’an, China, in 2010, 2013, and 2018, respectively. He is currently a Research Assistant Professor (RAP) with the Department of Computing, The Hong Kong Polytechnic University. His research interests include low-power wide-area networks, localization, and mobile computing. He is a member of ACM.

Yuanqing Zheng (Member, IEEE) received the B.S. degree in electrical engineering and the M.E. degree in communication and information system from Beijing Normal University, Beijing, China, in 2007 and 2010, respectively, and the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, in 2014. He is currently an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless networking and mobile computing, acoustic and RF sensing, and the Internet of Things (IoT). He is a member of ACM.

Tao Gu (Senior Member, IEEE) received the bachelor’s degree from the Huazhong University of Science and Technology, the M.Sc. degree from Nanyang Technological University, Singapore, and the Ph.D. degree in computer science from the National University of Singapore. He is currently an Associate Professor with the School of Computer Science and IT, RMIT University, Australia. His research interests include mobile computing, ubiquitous computing, wireless sensor networks, sensor data analytics, and the Internet of Things. He is a member of ACM.