SoftNB: A Fully Functional NB-IoT PHY for Various SDR Platforms

Jingze Zheng[†], Chaojie Gu[‡], Yuanchao Shu[‡], Xiuzhen Guo[‡], Shibo He[‡], Zhiguo Shi[†], Jiming Chen[‡§]
[†]College of Information Science & Electronic Engineering, Zhejiang University, China
[‡] College of Control Science and Engineering, Zhejiang University, China
[§] School of Automation, Hangzhou Dianzi University, China

***Abstract*—The design of Low Power Wide Area Network (LPWAN) protocols has attracted increasing attention in recent years, particularly within the LoRa research community. However, NB-IoT, another critical LPWAN technology, has not seen similar growth in its research community due to the lack of a functional and flexible software-defined radio (SDR) implementation. To address this gap, we present SoftNB, the first fully functional physical layer SDR implementation for NB-IoT. SoftNB conforms to the latest 3GPP standards and features an efficient and effective signal processing pipeline to mitigate time, frequency, and phase offsets during transmission and reception. Additionally, SoftNB is compatible with various SDR platforms, including USRP, HackRF One, and RTL-SDR Dongle. Extensive evaluations of SoftNB demonstrate its superior performance. Compared to the state-of-the-art baseline, SoftNB achieves an $8\times$ reduction in Block Error Rate (BLER) when the number of repetitions is set to 4 at a distance of 450 meters.**

***Index Terms*—NB-IoT, SDR, Physical Layer.**

I. Introduction

Recent years have witnessed notable advancements in Low Power Wide Area Network (LPWAN) technologies [1], such as LoRa [2], NB-IoT [3], Sigfox [4], and LTE-M [5]. LPWAN can cover a large geographical area due to its long-range communication capability. As a result, it has been widely adopted in a broad spectrum of Internet of Things (IoT) applications. As standards for these LPWAN technologies continue to evolve, this domain remains a vibrant arena of exploration for both industry and academia. Most of these studies leverage software-defined radios (SDRs) to access the physical layer (PHY) of a protocol to conduct research [6]. Since SDR bridges signals between the radio frequency and digital domains by implementing most of the network interface functionality in software (e.g., the physical and link layers), it offers developers the flexibility to modify or prototype this functionality as needed. For instance, researchers have utilized SDRs to empower LoRa networks with new features, including security enhancement [7], protocol development [8], concurrent decoding [9], open testbed [10], and C-RAN [11]. Furthermore, it paves the way for new applications such as vital monitoring, human activity sensing, localization, and backscattering.

This work was supported in part by the Key Research and Development Program of Zhejiang (No. 2024C01065), the National Science Fund of China under Grant No. (62302439, 62394344), and the Fundamental Research Funds for the Central Universities (226-2024-00004). Chaojie Gu is the corresponding author.

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

TABLE I: Comparison between SoftNB and related works.

Features	OAI[12]	Amarisoft[13]	Sonica[14]	**SoftNB**
Uplink	✔	✔	✔	✔
Downlink	✖	✖	✔	✔
3GPP Std.	Rel.13	Rel.13	Rel.17	Rel.17
Configurable	✖	✖	✔	✔
Open-source	✔	✖	Upon Req	✔
Calibration-free	✖	✖	✖	✔
Compatible w. Low-cost SDRs	✖	✖	✖	✔
Performance (NRep, Dist.)	N/A	N/A	N/A	6% BLER (4, 450 m)

Unfortunately, the NB-IoT research community does not enjoy the same level of prosperity as the LoRa community. As shown in Tab. I, while there exist some implementations of NB-IoT PHY for SDRs, they face challenges in terms of *functionality* and *flexibility*. Thus, in-situ deployment of NB-IoT PHY on SDRs is not available for rapid prototyping, testing, or verification.

- **Functionality.** OAI and Amarisoft support only uplink transmission from User Equipment (UE) to the base station (eNodeB), lacking support for downlink transmission from eNodeB to UE. Sonica necessitates synchronized clocks between eNodeB and UE, limiting its applicability. Additionally, OAI and Amarisoft only partially comply with the earliest 3GPP standard (Release 13), hampering both PHY and the upper-layer new design of NB-IoT.
- **Flexibility.** A large-scale NB-IoT network testbed requires the SDR implementation to be compatible with various classes of SDR platforms. However, Amarisoft is proprietary and necessitates dedicated devices, with the basic model costing $24,999. Sonica is tested on high-end USRP platforms (e.g., X300/310, B200/210) due to its requirement for clock synchronization between the UE and eNodeB. Similarly, OAI supports only USRP platforms (e.g., X310, B210, B205mini) and requires calibration with the Evolved Packet Core (EPC) before transmission. Consequently, building an NB-IoT network testbed with existing PHY implementations for development and research is prohibitively expensive.

These factors make it challenging to form a large-scale testbed for NB-IoT networks for developmental and research

purposes. To this end, in this paper, we present SoftNB, an open-source NB-IoT PHY solution. SoftNB is the first fully functional NB-IoT PHY supporting uplink and downlink communication following the latest 3GPP standard. Notably, SoftNB is open source and highly customizable, empowering users to tailor the transmission and reception chains of NB-IoT to their specific needs. Its versatility extends to deployment capabilities, as it is compatible with various commercial off-the-shelf (COTS) SDR platforms. This flexibility facilitates on-field testing and experimentation, which is crucial for advancing NB-IoT research and development. Despite the efforts required to align with the 3GPP standard, the signal is affected by hardware imperfections, environmental noise, and RF component noise. This leads to offsets in time, frequency, and phase. SoftNB achieves robust signal demodulation through the following design.

• **Signal arrival time estimation**: Time offset (TO) occurs when the signal arrival time is not accurately estimated. NB-IoT uses Timing Advance (TA) from the EPC that describes the signal propagation delay to estimate the signal arrival time. However, in practice, it is desirable to estimate signal arrival time without relying on the EPC. SoftNB achieves this by performing downlink (DL) synchronization using synchronization signals (§IV-A) and uplink (UL) synchronization based on the Cyclic Prefix (CP) (§IV-B). The proposed synchronization approach is highly tolerant of time offsets (§IV-C).

• **Frequency offset compensation**: Sampling clock frequency offset (SFO) arises due to oscillator mismatch between transceivers, leading to errors during fast Fourier transform and causing a notable degradation in communication performance. This issue is particularly pronounced in low-cost SDR platforms with unstable oscillators. To mitigate this, SoftNB introduces a frequency compensation algorithm based on the reference signal of NB-IoT (§V).

• **Phase offset estimation**: Phase offset (PO) refers to the angular difference between the ideal and measured phasors. NB-IoT adopts Quadrature Phase Shift Keying (QPSK) and Binary Phase Shift Keying (BPSK) in uplink and downlink channels, whose performance is sensitive to phase offset. However, during long-distance communication, phase offset is introduced by various factors such as channel conditions, antennas, amplifiers, and RF components of SDR devices. To address this challenge, SoftNB incorporates a phase de-rotation method (§VI-B) and Costas Loop (§VI-C) to estimate phase offset and enhance demodulation accuracy.

We implement and evaluate SoftNB on COTS SDR platforms, i.e., USRP N210 ($5,185 with UBX-40 as RF daughter board), HackRF One ($350), RTL-SDR ($33, RX only) in various cases including short/long distance and outdoor/indoor. Experimental results show that SoftNB achieves 4% Bit Error Rate (BER) and 6% Block Error Rate (BLER) when the transmission repetition is set as 4 at the distance of $450\,\mathrm{m}$, $8\times$ better than the state-of-the-art Sonica (PHY)[1]. Our key contributions are summarized as follows:

- To the best of our knowledge, we present the first-of-its-kind end-to-end implementation of a fully functional NB-IoT PHY. SoftNB can further empower the NB-IoT research community.
- We design an effective and robust signal processing pipeline to improve the transceiving performance of NB-IoT PHY channels and signals across different classes of SDR platforms.
- We extensively evaluate the performance of SoftNB under different conditions. The results show that SoftNB outperforms the state-of-the-art approaches.

The rest of this paper is organized as follows. §II introduces preliminary knowledge of NB-IoT. §III overviews the design. §IV to §VI present the design of addressing time, frequency, and phase offsets. § VII-B presents the implementation and experimental results under different conditions. §VIII reviews related work. Finally, §IX concludes this paper.

II. NB-IoT Primer

A. NB-IoT PHY

In an NB-IoT network, the User Equipment (UE), also known as the end node, establishes a connection with an eNodeB base station to communicate with the EPC. The UE is controlled through a series of ATtention (AT) commands. The uplink (UL) refers to transmissions initiated by the UE towards the eNodeB, while the downlink (DL) refers to transmissions sent from the eNodeB to the UE.

Modulation. NB-IoT adopts Single-Carrier Frequency Division Multiple Access (SC-FDMA) and Orthogonal Frequency Division Multiple Access (OFDMA) for UL and DL transmissions, respectively. QPSK and BPSK are adopted for symbol modulation and 16 Quadrature Amplitude Modulation (16QAM) is firstly introduced in Release 17 for symbol modulation in NB-IoT.

Bandwidth. The UL and DL of NB-IoT happen on different frequency bands. The UL of NB-IoT takes $180\,\mathrm{kHz}$ bandwidth in total, and supports $15\,\mathrm{kHz}$ or $3.75\,\mathrm{kHz}$ subcarrier spacing. The DL of NB-IoT occupies 12 subcarries of $15\,\mathrm{kHz}$ each, only supports a $15\,\mathrm{kHz}$ subcarrier spacing [15].

Operation modes. The NB-IoT aims to coexist with LTE carriers while maintaining the performance of the LTE. Thus, it supports three operation modes [16] as follows. *(1) In-band mode:* NB-IoT occupies a Physical Resource Block (PRB) of LTE network spectrum. *(2) Guard-band mode:* NB-IoT is deployed at the guard band of LTE that keeps LTE network uninfluenced. *(3) Stand alone mode*: NB-IoT signal occupies $180\,\mathrm{kHz}$ from $200\,\mathrm{kHz}$ carrier of the global system for mobile communications (GSM) system. In this case, NB-IoT has a $10\,\mathrm{kHz}$ guard band on both sides of the spectrum.

Framing. An NB-IoT frame comprises 10 subframes (SFs), with a total duration of $10\,\mathrm{ms}$. Fig. 1 illustrates the DL frame format, where each subframe can be divided into two slots, each containing 7 OFDM symbols. The UL frame format is the same as the DL if the UL adopts a $15\,\mathrm{kHz}$ spacing. If the

[1]Sonica does not have UL synchronization design in PHY. We apply our synchronization design to it for a fair comparison.

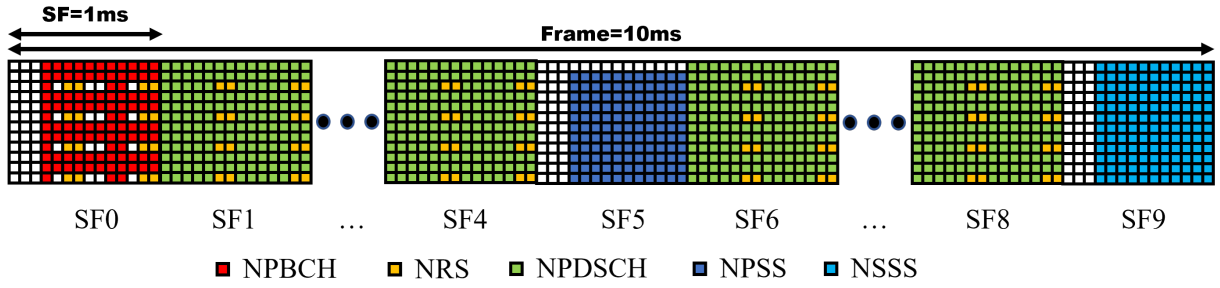


Fig. 1: NB-IoT Downlink frame for Guard-band/Stand-alone operation mode.

UL adopts a 3.75 kHz spacing, the radio frame format and the representation of the resource elements change accordingly.

B. NB-IoT Channels and Signals

NB-IoT inherits its essential channels and signals from LTE. A *channel* comprises a group of resource elements that convey information originating from higher layers (e.g., MAC). A *signal* is utilized by the physical layer but does not carry information from higher layers.

DL Channels and Signals. For DL, NB-IoT defines three physical channels: Narrowband Physical Downlink Shared Channel (NPDSCH), Narrowband Physical Broadcast Channel (NPBCH), and Narrowband Physical Downlink Control Channel (NPDCCH). The NPBCH contains essential information required by UE to receive further essential system information. The NPDSCH is used to transmit system information blocks and data. The NPDCCH is dedicated to transmitting control information from the network to the UEs. DL has three physical signals: Narrowband Primary Synchronization Signal (NPSS), Narrowband Secondary Synchronization Signal (NSSS), and Narrowband Reference Signal (NRS). The NPSS enables the UE to perform time and frequency synchronization, while the NSSS carries the cell identity (cell ID) information. The NRS, commonly known as "pilot," facilitates channel estimation in the frequency domain.

UL Channels and Signals. For UL, NB-IoT defines two physical channels: Narrowband Physical Uplink Shared Channel (NPUSCH) and Narrowband Physical Random Access Channel (NPRACH). The NPUSCH is dedicated to the transmission of data and control information, while the NPRACH is for transmitting the NB-IoT preamble, the first signal sent by the UE to request access to the network. Demodulation Reference Signal (DMRS) is the only signal defined in UL, also known as "uplink pilot," dedicated to channel estimation in the frequency domain for uplink transmissions.

III. SoftNB Overview

Besides supporting essential functions, e.g., UL/DL transmission, cell searching, SoftNB facilitates in-situ deployment and ensures compatibility with low-cost SDR platforms. Fig. 2 presents the overall transmission chain of SoftNB. Specifically, to achieve robust synchronization, SoftNB exploits the signal features of NPSS and CP to synchronize the DL and UL signals, respectively. To mitigate the impacts introduced by frequency offsets, SoftNB estimates these offsets in both DL and UL using NRS and DMRS, performing frequency compensation accordingly. Additionally, SoftNB achieves phase estimation with a phase demodulation module and a Costas loop in the UL. Compared with the design of Sonica or OAI, SoftNB performs phase de-rotation in time domain before FFT thus enhancing the resolution of each symbol. In addition, SoftNB uses Costas Loop for phase compensation which shows better performance in SDR devices. To support Release 17, SoftNB also supports 16QAM in symbol modulation.

IV. Synchronization

SoftNB achieves DL synchronization based on NPSS and UL synchronization based on CP. As a system function in PHY, SoftNB needs to synchronize each UL signal respectively, while other SDR platforms (e.g., Sonica) allocate time resource for UE so that UL synchronization based on PHY is unnecessary.

A. DL synchronization with NPSS

NPSS, based on Zadoff-Chu (ZC) sequence [17], is designed for time synchronization for UEs in DL communication. The ZC sequence, denoted by $c_{l,k}$, is expressed as:

$$c_{l,k} = a_l e^{-\frac{\pi u k(k+1)}{11}}, \tag{1}$$

where $l = \{3, 4, 5, ..., 13\}$ is the index of OFDM symbols within the subframe, $k = \{0, 1, 2, ..., 10\}$ is the index of the subcarriers, $\{a_3, a_4, a_5..., a_{13}\} = \{1, 1, 1, 1, -1, -1, 1, 1, 1, -1, 1\}$ and $u = 5$ for the ZC root sequence index.

To accurately achieve synchronization, it is significant to exploit the feature of NPSS. It is desirable to adopt Maximum-likelihood (ML) algorithms due to their high efficiency for this task [18]. Both auto-correlation and cross-correlation methods are feasible in the processing of NPSS. Although the cross-correlation technique is optimal in ML sense, it requires a known sequence at the receiver side and is computationally intensive [3]. Therefore, we select auto-correlation for SoftNB as NPSS is based on ZC sequence which has a self-correlated feature hardly influenced by frequency/phase offset.

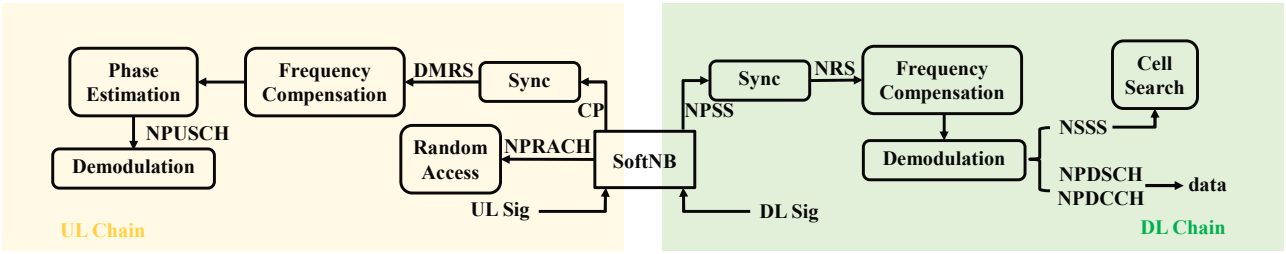
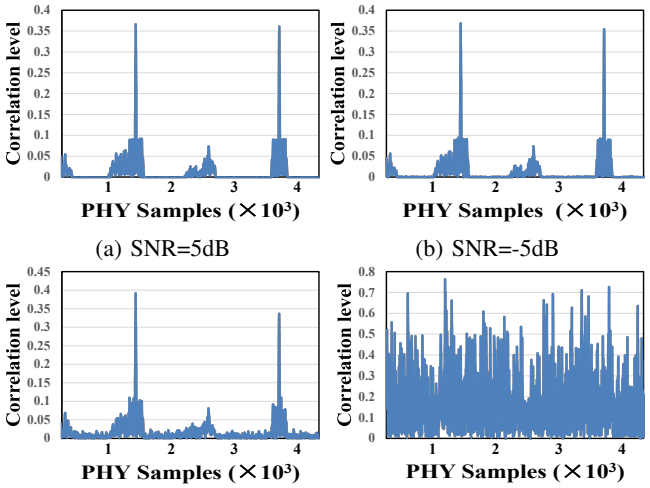


Fig. 2: Overall transmission chain of SoftNB.



(a) SNR=5dB (b) SNR=-5dB

(c) SNR=-15dB (d) SNR=-25dB

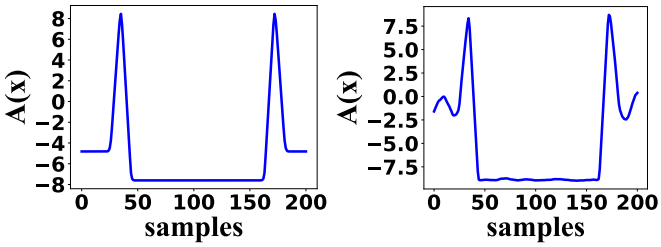
Fig. 3: Synchronization performances with different SNRs.

To reduce the computation overload, we downsample the original sequence to 240 kHz. The auto-correlation is given by:

$$A(x) = \left| \sum_{i=1}^{NS-l} c[i] \sum_{j=1}^{len} s[x+j]\overline{s}[x+j+lag] \right|, \quad (2)$$

where $A(x)$ denotes the auto-correlation level at sample x. $NS = 14 - 3 = 11$ for that the first 3 symbols of SF5 are empty, $len = 17$ denotes the length of one symbol and CP after downsampling. $\{c_1, c_2, c_3, ..., c_{10}, c_{11}\}$={1,1,1,1,-1,-1,1,1,1,-1,1} as defined in Release 13 [15]. s denotes the sequence after resampling and lag is the lag time of auto-correlation and we set it as 51 samples (3 SFs) in our algorithm.

We analyze a DL waveform with auto-correlation at different Signal Noise Ratios (SNRs), of which the result is shown in Fig. 3. The peaks of auto-correlation present the location of the start of NPSS. It is found that auto-correlation performs well at an SNR of -15dB. Note that NB-IoT only requires an SNR of -11.8dB for accurate demodulation [19]. Thus, SoftNB can precisely locate NPSS through the auto-correlation algorithm. And its performance remains stable while the SNR is better than -15dB.



(a) Synchronization of standard signal. (b) Synchronization of signal received by HackRF One.

Fig. 4: Synchronization based on CP.

B. UL synchronization based on CP

CP is introduced to resist Inter-Symbol Interface (ISI) and Inter-Channel Interface (ICI). It copies the end of a modulated symbol to its start. We notice CP can be used for auto-correlation-based synchronization [20] in the UL case. To illustrate, the synchronization algorithm is defined as:

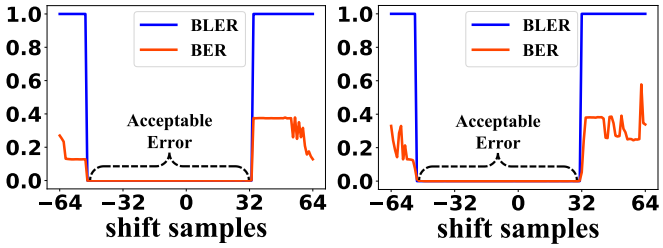
$$A(x) = Re(\sum_{j=1}^{len} s[x+j]\overline{s}[x+j+lag]), \quad (3)$$

where $len = 9$ denotes the length of normal CP, $lag = 128$ represents the length of a symbol. Fig. 4 shows the results of the proposed auto-correlation approach for a local-generated standard signal and a signal received by an SDR receiver, respectively. We can find that the auto-correlation coefficient climbs when the convolution window moves to the CPs. Finding the peak of auto-correlation, we can distinguish the pattern of CP from samples, thus determining the start of every symbol.

C. Tolerance of synchronization error

For UL cases, CP-based synchronization results may deviate from the real situation as the length of CP is short.

For standard signals, auto-correlation can always locate CP accurately. However, for signals captured by SDR devices, the result of synchronization may be inaccurate due to noise, multipath effect, or other channel influences. Nevertheless, SoftNB exhibits a high tolerance to error in synchronization. Taking a signal pattern with 128 slots (repeat for 4 times) as an example, we manually introduced error in synchronization and recorded the BER and BLER for both standard signal and received signal in Fig. 5. It can be found that a certain



(a) Standard signal. (b) Received signal.

Fig. 5: Tolerance to synchronization error.

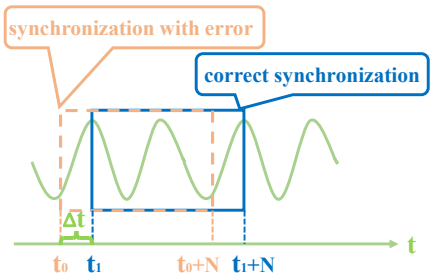


Fig. 6: Results of synchronization.

range synchronization error does not affect the process of demodulation and decoding. SoftNB then removes the pattern of CP according to the output of synchronization for further demodulation.

According to the expression of FFT, the error introduced by time synchronization can be partly neutralized by other samples in the frame window after synchronization through the process of series summing. The process of Fourier transformation is expressed as:

$$X[k] = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}kn} x[n], \tag{4}$$

where $X[k]$ represents the intensity of the signal at the frequency mapped to FFT bin k, and $N = 128$ is the length of each symbol. We demonstrate the result of correct synchronization and wrong synchronization in Fig. 6.

When we adopt FFT to perform QPSK demodulation, we can get

$$\begin{aligned} X[k] &= \sum_{n=t_1}^{t_1+N} e^{-j\frac{2\pi}{N}kn} x[n], \\ \hat{X}[k] &= \sum_{n=t_0}^{t_0+N} e^{-j\frac{2\pi}{N}kn} x[n], \end{aligned} \tag{5}$$

in which $\hat{X}$ represents the symbols demodulated with the wrong synchronization.

As the rotation of symbols can be corrected through NRS, so we define the error of each symbol as:

$$\begin{aligned} \Delta X[k] &= X[k] - \hat{X}[k] e^{j\frac{2\pi}{N}k\Delta t}, \\ &= \sum_{n=t_0+N}^{t_1+N-1} e^{-j\frac{2\pi}{N}kn} x[n] - \sum_{n=t_0}^{t_1-1} e^{-j\frac{2\pi}{N}k(n-\Delta t)} x[n], \\ &= e^{-j\frac{2\pi}{N}k(n-\Delta t)} \sum_{n=0}^{\Delta t-1} (e^{-j\frac{2\pi}{N}k\Delta t} x[N+n] - x[n]). \end{aligned} \tag{6}$$

Considering that the phase error can be corrected by NRS, we focus on analyzing the norm of $\Delta X[k]$:

$$\left|\Delta X[k]\right| = \left| \sum_{n=0}^{\Delta t-1} (e^{-j\frac{2\pi}{N}k\Delta t} x[N+n] - x[n]) \right|. \tag{7}$$

where Δt, denoting the offset of synchronization, is less than the length of CP (10 samples for extended and 9 samples for normal), $x[N+n] = x[n]$, Eq. 7 can be simplified to

$$\left|\Delta X[k]\right| = \left| \sum_{n=0}^{\Delta t-1} x[n](e^{-j\frac{2\pi}{N}k\Delta t} - 1) \right| \triangleq o(\Delta t), \tag{8}$$

where $|\Delta X[k]|$ positively correlated with Δt. Thus, we conclude that the error increases when the synchronization offset Δt rises. However, as the signal gathers at low frequency, k is always an integer close to 0. With the length of CP is defined as L, $|\Delta X[k]|$ is a small amount when $\Delta t < L$. If we consider the cases of $\Delta t < 0$ and $\Delta t > L$, Eq. 8 can be expressed as:

$$\left|X[k]\right| = \begin{cases} |\sum_{n=1}^{\Delta t-L} (e^{-j\frac{2\pi k\Delta t}{N}} x[N+n] - x[n])| + o(L), \Delta t > L; \\ |\sum_{n=0}^{-\Delta t-1} (e^{j\frac{2\pi}{N}k\Delta t} x[n] - x[n+N])|, \Delta t < 0; \\ o(\Delta t), 0 < \Delta t \leq L. \end{cases} \tag{9}$$

This also explains that the acceptable error region is not symmetrical. For the case that $\Delta t > 0$, the synchronization error is partly neutralized by CP so $o(\Delta t)$ is a small amount compared with another item of $|\Delta X[k]|$. Therefore, the case that $\Delta t < 0$ is more acceptable for the system, which perfectly matches the results of experiments.

V. Frequency Compensation

For SDR devices, especially low-cost SDR devices, SFO caused by unstable oscillators is a common and critical issue in long-distance communication. Although we expect the transmitter and receiver to produce the same carrier frequency, the carrier frequency can be difficult to maintain due to the inherent physical characteristics of the oscillator. We use f_c and f_c' to denote the frequency of the transmitter and receiver, respectively. The frequency offset f_o is $f_c - f_c'$.

We set $\epsilon = \frac{f_o}{\Delta f}$, where Δf presents the sub-carrier spacing, $\Delta f = 15$kHz for DL and $\Delta f = 15$kHz/3.75kHz for UL. A frequency offset of ϵ can cause a phase rotation of $2\pi\epsilon$ [21], so the estimated frequency offset $\hat{\epsilon}$ is expressed as:

$$\hat{\epsilon} = \frac{1}{2\pi(L_{CP}+N)} \arg\{ \sum_{n=1}^{L_{CP}+N} y^*[n]x[n] \}, \tag{10}$$

where $L_{CP} = 9$ presents the length of CP, $N = 128$ presents the length of symbol, $y[n]$ is the reference symbol received and $x[n]$ is the symbols transmitted. Based on the assumption that frequency offset does not change significantly over a NB-IoT slot, we estimate the frequency offset with NRS and DMRS, the reference symbol for DL/UL cases. We perform frequency compensation for the received signal as:

$$y'[n] = y[n]e^{\frac{-2\pi\Delta f\hat{\epsilon}n}{f_s}}, \tag{11}$$

in which we select $f_s = 1.92\,\text{MHz}$ as the sample rate in our design.

VI. Phase Estimation

The UL chain of NB-IoT is shown in Fig. 7. For UL cases, with the design of phase rotation, QPSK symbols are dispersed in the constellation diagram with a lower phase resolution. Therefore, we design a phase de-rotation algorithm and adopt Costas Loop to perform phase estimation.

A. UL Signal Generation

To test the performance of SoftNB in UL, we need to generate a standard UL signal. In this section, we explain the procedure of UL signal generation and emphasize the phase rotation part to help us understand the design of the phase estimation of SoftNB. To reduce power consumption, Release 13 stipulated that the modulation scheme for the NB-IoT UL is Single-carrier Frequency Division Multiple Access (SC-FDMA) [22]. According to Release 13, the baseband signal is generated as:

$$s_k(t) = a_k \cdot e^{j\pi\Delta f(t-L_{cp}T_s)} \cdot e^{j\phi}, \tag{12}$$

in which

$$\begin{aligned} &0 \le t \le (L_{cp}+N)Ts \\ &N = 128 \\ &T_s = \frac{1s}{1.92M} = 5.2083\times10^{-7}s \\ &L_{cp} = \begin{cases} 9, & k\ mod\ 7 = 0 \\ 10, & others \end{cases} \\ &\Delta f = \begin{cases} 3.75\,kHz \\ 15\,kHz \end{cases}. \end{aligned} \tag{13}$$

In Eq. 12, $a_k = \{\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j, \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j, -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j, -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j\}$ is the QPSK symbol of index k. L_{cp} and N denote the number of samples of symbols and CPs (extended CP or normal CP), respectively. T_s is sampling time with sampling rate of $1.92\,\text{MHz}$ and Δf is the sub-carrier spacing defined in protocol. Additionally, ϕ, which denotes the rotation angle of a_k, is also defined in the protocol and known by the transmitter and receiver. ϕ is expressed as:

$$\phi(l) = \rho(l\ mod\ 2) + \hat{\phi}(l-1), \tag{14}$$

in which

$$\begin{aligned} \rho &= \begin{cases} \frac{\pi}{2} & ,BPSK \\ \frac{\pi}{4} & ,QPSK \end{cases} \\ \hat{\phi}(l) &= \begin{cases} 0 & ,l = 0 \\ \hat{\phi}(l-1) + \pi\Delta f(N+N_{cp})T_s & ,l > 0 \end{cases} \end{aligned} \tag{15}$$

where l is used for counting, zeroing at the beginning of the transmission, and increasing as the number of symbols increases during the transmission.

B. Phase de-rotation

For the receiver side, we perform phase de-rotation after synchronization and frequency compensation. The procedure of phase de-rotation is expressed as:

$$\begin{aligned} f_r[i][j] &= \begin{cases} p[i] & ;j = f_t \\ 0 & ;else \end{cases} ;0 \le j \le L_S, \\ p[i][j] &= \sum_{t=0}^{L_S} f_r[i][j]e^{j2\pi\frac{tj}{L_S}}, \\ \widetilde{s}[i] &= \frac{s[i]}{p[i]}, \end{aligned} \tag{16}$$

where $f_r[i]$ represents the phase shift of symbol whose index is i, $L_S = 128$ presents the length of symbol, $p[i]$ is the phase information recorded and the FFT bin f_t is set as 123 according to the spectrum of baseband signal. p_r denotes the rotation sequence in time domain and $\widetilde{s}[i]$ represents the symbol sequence after phase de-rotation. We perform phase de-rotation in time domain before demodulation thus enhancing the resolution of each symbol.

C. Costas Loop

After phase de-rotation, we get a sequence modulated with QPSK of which each symbol possesses 128 samples. Fig. 8a shows the result of directly applying demodulation.

The process of frequency/phase offset and channel noise is important for long-distance signal reception. We adopt Costas Loop, which consists of a controlled Oscillator (VCO), Loop Filter (LPF), and phase detector, for frequency and phase synchronization. The specific form of Costas Loop is shown in Fig. 9.

The input signal is multiplied by the inverting and quadrature signals output by the local oscillator respectively to obtain the signal $s_I(t)$ containing the error information of the inverting component and the signal $s_Q(t)$ containing the error information of the quadrature component. With operational efficiency and accuracy in mind, we replace the traditional multiplication phase detector with a judgment-oriented phase detector. The output of the phase detector is expressed as:

$$V_D(t) = s_Q(t)Sign(s_I(t)) - s_I(t)Sign(s_Q(t)), \tag{17}$$

in which $Sign$ denotes a symbolic function. The loop filter determines the transmission characteristics of the whole loop, and the design of the parameters of the digital loop filter is also the focus and difficulty of the Costas Loop. The system function of LPF is expressed as:

$$F(Z) = C_1 + \frac{C_2 z^{-1}}{1 - z^{-1}}. \tag{18}$$

For iteration convenience, we replace C_1, C_2 with an integrated gain G_{lf}. The expression of the loop filter can be rewritten as:

$$O_{lf}(t+1) = O_{lf}(t+1) + G_{lf}V_D(t). \tag{19}$$

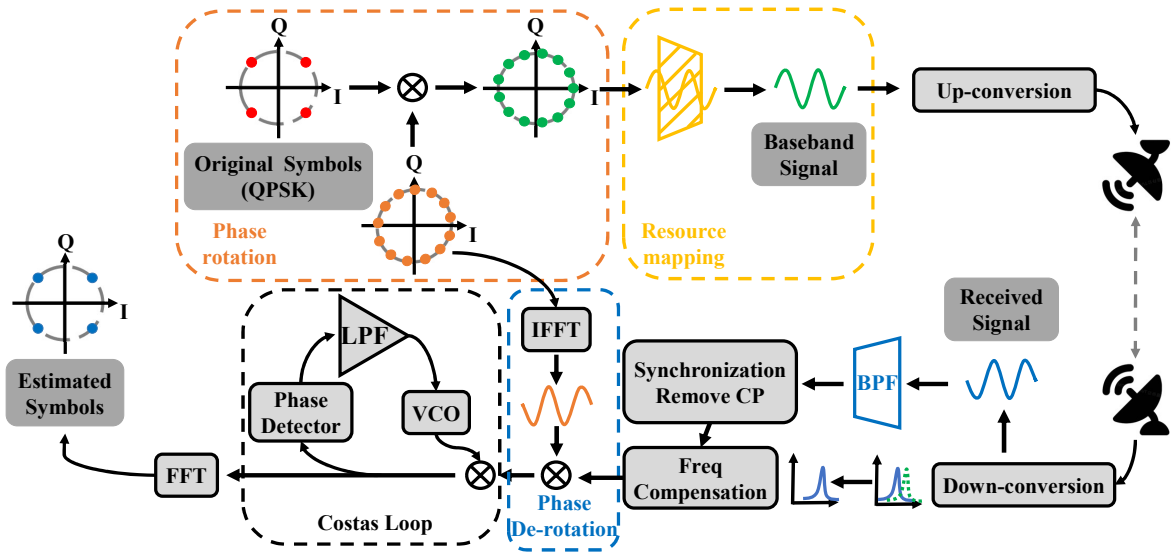
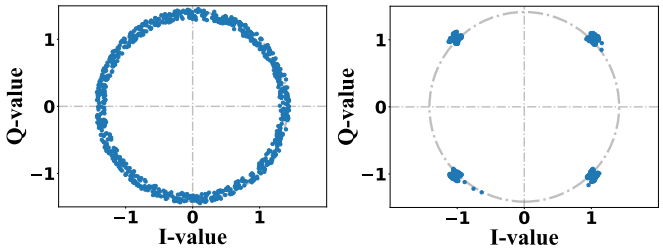


Fig. 7: UL process of NB-IoT.



(a) Demodulation without Costas loop. (b) Demodulation with Costas loop.

Fig. 8: The result of demodulation.

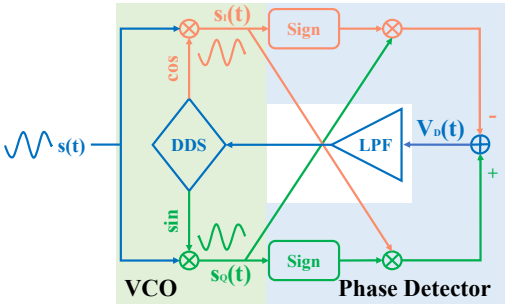


Fig. 9: Digital Costas Loop.

For the VCO module, we alternate it with a Direct Digital Synthesizer (DDS), which generates a signal with a phase of O_{lf}. According to the output of Costas Loop, we can get a sequence ready for QPSK demodulation, as demonstrated in Fig. 8b. It is noteworthy that symbols are rotated as the uncertainty of the original phase, so we need to perform phase compensation with reference symbols.

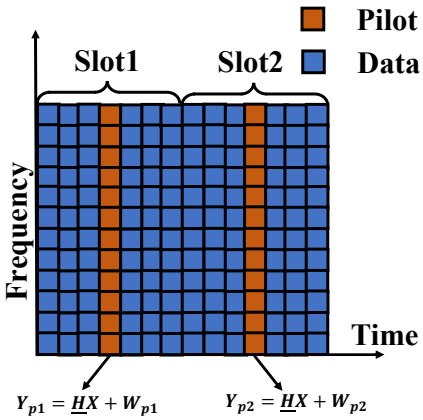


Fig. 10: Channel estimation one subframe, with M = 12.

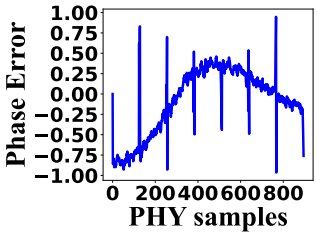
D. The advantage of Costas Loop

Compared with MMSE [23] adopted in other SDR systems (e.g., Sonica [24]) supporting NB-IoT, Costas Loop inherently has a better performance in phase synchronization and demodulation.

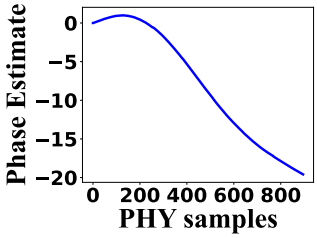
According to [23], MMSE equalizer aims to find the diagonal equalization matrix $\underline{\mathbf{Q}}$ which minimizes the error function $e = E\{||\mathbf{X}_n - \hat{\mathbf{X}}_n||^2\}$, where $E\{.\}$ is s the mathematical expectation, while $\hat{\mathbf{X}}_n = \underline{Q}\mathbf{Z}_n$ denoting the estimated transmitted symbols under the MMSE criterion. The minimization of e leads to:

$$\underline{\mathbf{Q}} = \underline{\mathbf{H}}_n^H (\underline{\mathbf{H}}_n \underline{\mathbf{H}}_n^H + (E\{\underline{\mathbf{X}}_n \underline{\mathbf{X}}_n^H\}))^{-1} \sigma_W^2)^{-1}. \quad (20)$$

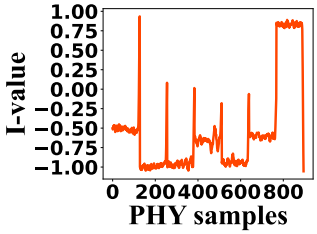
The MMSE equalizer does not tend to amplify the noise, due to the contribution of the term $E\{\underline{\mathbf{X}}_n \underline{\mathbf{X}}_n^H\}))^{-1}\sigma_W^2)$. To estimate the noise σ_W^2 for NB-IoT, we take one subframe containing 12 symbols as an example.



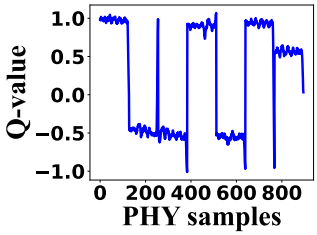
(a) Phase error of a slot with 7 symbols.



(b) Phase estimated of a slot with 7 symbols.



(c) In-phase waveform after Costas Loop.



(d) Quadrature waveform after Costas Loop.

Fig. 11: Analysis of Costas Loop.

We can express the estimated noise variance as:

$$\sigma_W^2 = \frac{1}{2M}\sum_{m=0}^{M-1}|Y_m^-|^2, \qquad \mathbf{Y}^- = \mathbf{Y}_{p1} - \mathbf{Y}_{p2}. \tag{21}$$

Therefore, if we take a subframe as a unit, the estimated $\underline{\mathbf{Q}}$ will function in 14 symbols accordingly. However, for SDR systems, especially those implemented on cheap SDR devices, the BER is mainly caused by the frequency and phase offset rather than the channel noise. For example, the fluctuation of the sampling rate can cause Carrier Frequency Offset (CFO), and the multipath effects can lead to Carrier Phase Offset (CPO). Costas Loop shows better performances in processing these two offsets. The phase error estimated by Costas Loop is shown in Fig. 11a. The phase error has fluctuations in a small range within 128 samples of a symbol. Therefore, if we adopt MMSE to estimate the channel, the details in a symbol will be lost thus introducing errors. The signal after the processing of Costas Loop is shown in Fig. 11b. It is noteworthy that the phase estimated represents the phase of the carrier rather than the waveform containing QPSK symbols. According to Fig. 11c and Fig. 11d representing In-phase symbols and Quadrature symbols, respectively, we can identify the QPSK symbols and bit stream easily.

VII. Implementation and Evaluation

A. Implementation

Transmitter. Fig. 12a presents the prototype of the SoftNB transmitter. The transmitter consists of two parts, a USRP N210 equipped with UBX-40 RF [25] daughterboard and an ASUS laptop (i9-14900HX, 2.2GHz, 16G RAM) as signal source and controller. As the accuracy of the transmitter is more significant than the receiver, we use USRP N210 to transmit the NB signal. The USRP uses an omnidirectional antenna with 5dB gain, of which the operation spectrum is 40MHz-2000MHz. While running, the laptop controls USRP with GNURadio, and USRP transmits the data stream from the laptop.

(a) USRP N210 as transmitter.

(b) HackRF One as receiver.

(c) RTL Dongle as receiver.

Fig. 12: SoftNB is compatible with different SDR platforms.

Receiver. Fig. 12b and Fig. 12c presents HackRF One and RTL Dongle respectively. For long-distance evaluations, we installed an RF signal amplifier with a gain of 20 dB on the receiver of which the operating band is 50MHz-6GHz. We use an SMA telescopic antenna for HackRF One and an omnidirectional antenna for RTL Dongle.

Cost. While SoftNB is a software design, it needs to be deployed on SDR devices to create an NB-IoT network. As previously mentioned, the SoftNB transmitter utilizes a USRP N210 paired with a UBX-40 daughterboard, costing $6,100. For receivers, we can employ USRP, HackRF One ($110), or RTL Dongle ($14). Given that an NB-IoT network comprises numerous UEs and relatively few eNodeBs, SoftNB significantly reduces the cost of building an NB-IoT network based on SDR platforms.

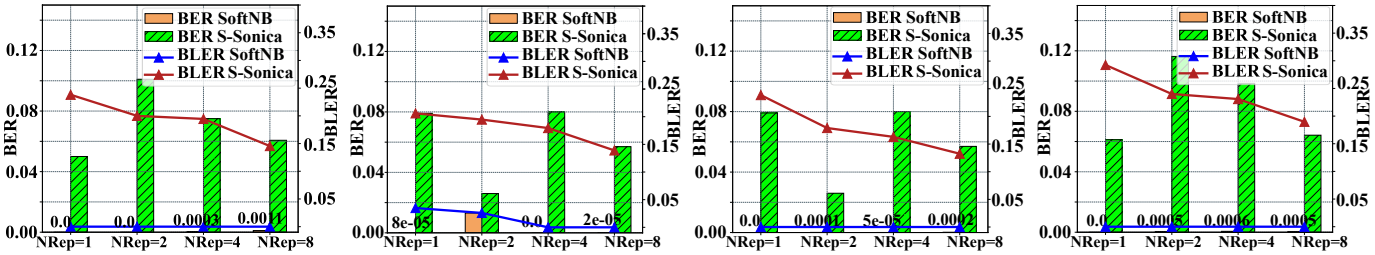
B. Evaluation

In this section, we conduct extensive experiments in various settings to evaluate the performance of SoftNB.

Baseline. Since Sonica does not support synchronization in the PHY layer, we apply our signal synchronization design to it for a fair comparison. For channel estimation, we adopt the MMSE approach used by Sonica [24]. We refer to the baseline as *S-Sonica*. Note that NB-IoT adopts the signal repetition scheme to improve its transmission accuracy [26]. Thus, we evaluate the baseline and SoftNB with different numbers of repetitions.

Metrics. We evaluate the performance of SoftNB using the following metrics.

- *Bit Error Rate (BER)* is the ratio of the number of bits received incorrectly to the total number of bits transmitted.
- *Block Error Rate (BLER)* is the ratio of incorrectly received blocks to the total number of transmitted blocks. In NB-IoT, the Block is defined in higher Layers. In our experiments, a block consists of 16 slots. A block is considered incorrectly received if any repetition within the block fails to achieve a BER of 0.



(a) Receiver indoor (location 1). (b) Receiver at F4 (location 2). (c) Receiver at F3 (location 3). (d) Receiver at F2 (location 4).

Fig. 13: Indoor experiments results.

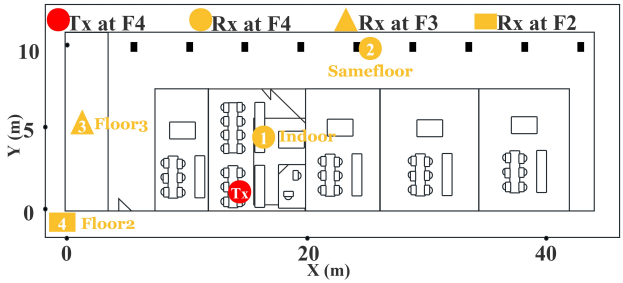


Fig. 14: Indoor experiments deployment.

1) Indoor experiments: **Setup.** The performance of SoftNB is assessed within a four-story concrete building. The floor plan of the fourth floor is depicted in Fig. 14. The TX power of the transmitter is 20 dBm. For the receiver, the Radio Frequency Gain (RF Gain), Intermediate Frequency Gain (IF Gain), Base Band Gain (BB Gain) are set as 15 dB, 35 dB, 15 dB, respectively. The carrier frequency is set as 1,800 MHz, belonging to the UL spectrum of NB-IoT. We fix the transmitter on the fourth floor and carry the receiver to different locations. For each location, we set the repetition time as 1, 2, 4, and 8. We set every transmitting block with a length of 224 bits. Each transmitting block is the output of Cyclic Redundancy Checking (CRC), channel coding, rate matching, and scrambling with an input of a randomly generated codeword sequence.

Results. The BERs and BLERs at each location are shown in Fig. 13. The BLER decreases as the number of repetitions increases. Compared to the baseline, SoftNB achieves close to 0 BLERs at all locations when the number of repetitions is 4. In contrast, the baseline method does not achieve a BLER of zero even with 8 repetitions. Therefore, SoftNB can significantly reduce the number of repetitions needed, thereby improving transceiving efficiency.

2) Outdoor experiments: **Setup.** To further evaluate the remote communication performance of SoftNB, we conduct outdoor experiments. We evaluate SoftNB and the baseline at playgrounds (Fig. 15a) and road (Fig. 15b) and considered different communication distances. According to indoor experimental results, the BLER of SoftNB is close to 0 when the number of repetitions is 4. Thus, we set the number of

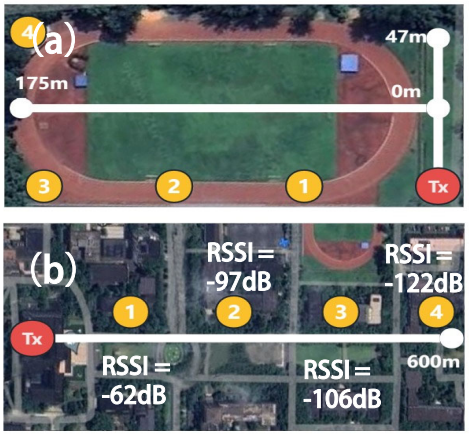


Fig. 15: Outdoor experiments setups: (a) Playground deployment (b) Road deployment. (Image credit: Google Map)

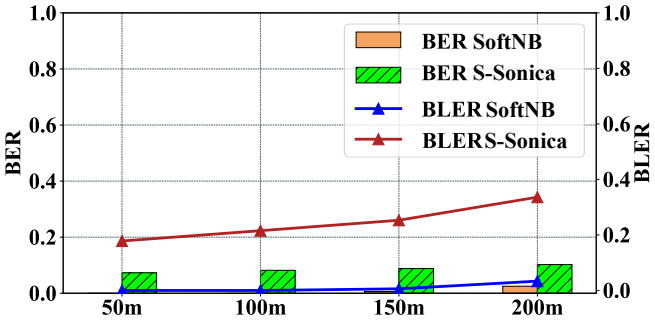


Fig. 16: Playground experiments results.

repetitions as 4 for outdoor experiments. We use HackRF one as the receiver and install a 20 dB gain amplifier. We set the TX power of the the transmitter to 20 dB, and place the receiver at a different distance from the transmitter.

For the experiments at the playground, as shown in Fig. 15b, we fix the position of the transmitter and carry the receivers to different locations, which are 50 m, 100 m, 150 m, and 200 m away from the transmitter, respectively. For road cases, we deploy transmitter and receiver as shown in Fig. 15d, we set four distances including 150 m, 300 m, 450 m and 550 m.

Results. Fig. 16 presents the performance of SoftNB at

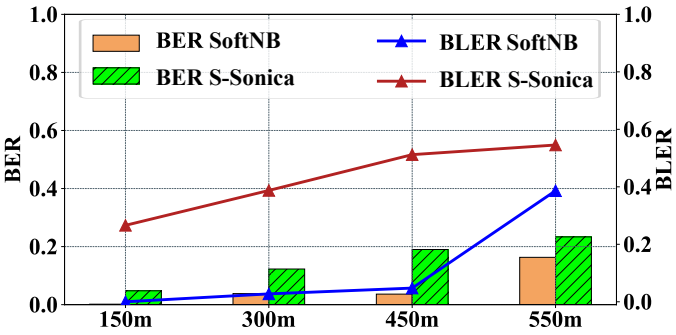


Fig. 17: Road experiments results.

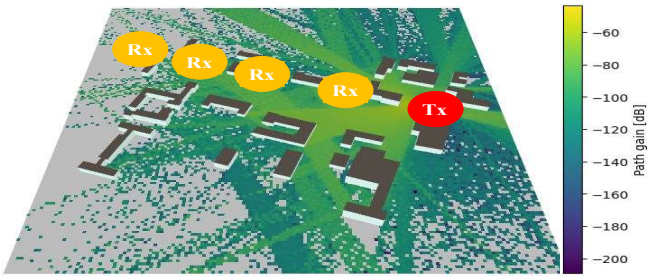


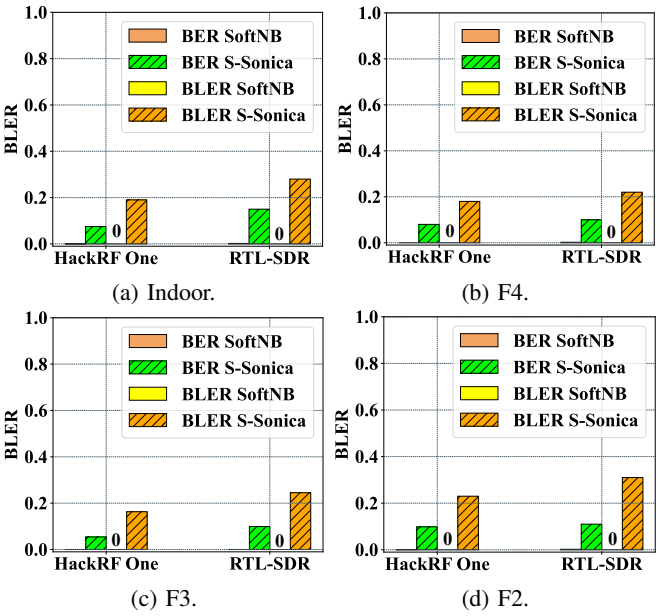
Fig. 18: Simulate with Sionna Ray Tracing. (The map is exported by Blender and OpenStreetMap)

different transmission distances. As we can see, the BER and BLER of SoftNB are always lower than the baseline. Furthermore, the BLER of SoftNB keeps 0 at 50 m and 100 m. At 150 m, BER and BLER show a winding upward trend with the distance. However, the BER and BLER of SoftNB are very low despite their climb. As for the performance on the road shown in Fig. 17, we can discover that the accuracy of SoftNB drops when the distance reaches 550 m while the system functions well within 500 m. When the distance of communication reaches 550 m, the performance of SoftNB and baseline both drops significantly.

To understand the performance of SoftNB when the distance between the transceiver pair is 550 m, we emulate the measurements with Sionna RT [27], a 3D ray tracing tool for radio wave propagation analysis. As shown in Fig. 18, the Received Signal Strength Indicator (RSSI) decreases as the distance increases due to obstacles along the path. The simulation results in Fig. 18 approximately match the onsite measurement results. At a distance of 550 m, the signal attenuates significantly, leading to high BER and BLER, which is consistent with our experimental findings.

C. RTL-SDR Dongle as receiver

To reduce system costs and enable large-scale deployments, SoftNB supports low-cost SDR devices (e.g., RTL-SDR Dongle) as receivers. Following the indoor environment settings, we conducted side-by-side experiments to compare the receiving performance of HackRF One and RTL-SDR Dongle. As shown in Fig. 19, for the baseline, both BER and BLER increase when replacing HackRF One with RTL-SDR



(a) Indoor. (b) F4.

(c) F3. (d) F2.

Fig. 19: The performance of HackRF/RTL.

Dongle. This is because the RTL-SDR Dongle uses low-cost RF components that are less stable. However, we find that SoftNB achieves consistent performance on both HackRF One and RTL-SDR Dongle due to our dedicated design.

VIII. Related Work

SDR Systems. To perform in-depth debugging of a wireless network protocol, it is common practice to use an SDR platform to build a testing network. For example, researchers have developed PHY implementations for LoRa [28], [29], [30], [31], [32], [33] and LTE [34]. In recent years, there have been some NB-IoT PHY implementations, such as OAI [35], Amarisoft [13], and Sonica [24].

NB-IoT. The main research regarding NB-IoT focuses on resource scheduling and power consumption. Yu et al. present an algorithm for the optimization of UL resource allocation in NB-IoT [36]. Yang et al. design NBScope, a diagnostic tool that supports fine-grained fusion of power and protocol traces [37]. Sultania et al. offer a simulator for the power consumption of NB-IoT [38].

IX. Conclusion

We present SoftNB, a fully functional NB-IoT PHY design for various SDR platforms, demonstrating high transmission accuracy over long distances and compatibility with low-cost SDR devices. We implement SoftNB and evaluate it on different SDR platforms, including USRP N210, HackRF One, and RTL-SDR Dongle. SoftNB outperforms the state-of-the-art baseline (Sonica) in both BER and BLER, and supports more affordable SDR devices such as HackRF One and RTL-SDR Dongle. By achieving this, SoftNB holds great promise for advancing NB-IoT research.

REFERENCES

[1] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[2] O. Alliance, "Lora." https://lora-alliance.org/.

[3] M. Kanj, V. Savaux, and M. Le Guen, "A tutorial on nb-iot physical layer design," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.

[4] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.

[5] S. R. Borkar, "Long-term evolution for machines (lte-m)," in *LPWAN Technologies for IoT and M2M Applications* (B. S. Chaudhari and M. Zennaro, eds.), pp. 145–166, Academic Press, 2020.

[6] H. Zamat and C. R. Nassar, "Introducing software defined radio to 4g wireless: Necessity, advantage, and impediment," *Journal of Communications and Networks*, vol. 4, no. 4, pp. 1–7, 2002.

[7] C. Gu, L. Jiang, R. Tan, M. Li, and J. Huang, "Attack-aware synchronization-free data timestamping in lorawan," *ACM Trans. Sen. Netw.*, vol. 18, oct 2021.

[8] A. Gamage, J. Liando, C. Gu, R. Tan, M. Li, and O. Seller, "Lmac: Efficient carrier-sense multiple access for lora," vol. 19, feb 2023.

[9] X. Xia, N. Hou, Y. Zheng, and T. Gu, "Pcube: scaling lora concurrent transmissions with reception diversities," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, (New York, NY, USA), p. 670–683, Association for Computing Machinery, 2021.

[10] M. Mishra, D. Koch, M. O. Shahid, B. Krishnaswamy, K. Chintalapudi, and S. Banerjee, "OpenLoRa: Validating LoRa implementations through an extensible and open-sourced framework," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, (Boston, MA), pp. 1165–1183, USENIX Association, apr 2023.

[11] J. Liu, W. Xu, S. Jha, and W. Hu, "Nephalai: towards lpwan c-ran with physical layer compression," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[12] O. S. Alliance, "Openairinterface." https://openairinterface.org/.

[13] Amarisoft, "Amarisoft." https://www.amarisoft.com/.

[14] S. L. Boyan Ding, Junhao Zhao, "Sonica (software-defined open nb-iot research platform)." https://dboyan.github.io/sonica-doc/.

[15] 3GPP, "Evolved universal terrestrial radio access (e-utra); physical layer procedures," Technical Specification (TS) 36.213, 3rd Generation Partnership Project (3GPP), 12 2021. Version 13.9.0.

[16] 3GPP, "Evolved universal terrestrial radio access (e-utra); user equipment (ue) radio transmission and reception," Technical Specification (TS) 36.101, 3rd Generation Partnership Project (3GPP), 01 2021. Version 13.9.0.

[17] R. Boisguene, S.-C. Tseng, C.-W. Huang, and P. Lin, "A survey on nb-iot downlink scheduling: Issues and potential solutions," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 547–551, 2017.

[18] H. Kroll, M. Korb, B. Weber, S. Willi, and Q. Huang, "Maximum-likelihood detection for energy-efficient timing acquisition in nb-iot," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–5, 2017.

[19] R. Vaibhav and T. S. Reddy, "Sample calculation of link power budget and effective snr in nb-iot," in *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, pp. 30–32, 2018.

[20] W. Guo, S. Hu, and S. Li, "Cyclic prefix-based time and frequency synchronization for td-lte downlink transmission," in *2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, 2014.

[21] F. Wang and Y. Zhu, "An efficient cfo estimation algorithm for the downlink of 3gpp-lte," in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, 2011.

[22] H. G. Myung, J. Lim, and D. J. Goodman, "Single carrier fdma for uplink wireless transmission," *IEEE Vehicular Technology Magazine*, vol. 1, no. 3, pp. 30–38, 2006.

[23] V. Savaux, H. Dembélé, and M. Kanj, "Uplink channel estimation and equalization in nb-iot system," in *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 167–174, 2019.

[24] B. Ding, J. Zhao, Z. Tan, and S. Lu, "Sonica: an open-source nb-iot prototyping platform," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, (New York, NY, USA), p. 868–870, Association for Computing Machinery, 2022.

[25] D. Stanko, G. Sommerkorn, A. Ihlow, and G. del Galdo, "Enable software-defined radios for real-time mimo channel sounding," in *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–5, 2021.

[26] R. Harwahyu, R.-G. Cheng, W.-J. Tsai, J.-K. Hwang, and G. Bianchi, "Repetitions versus retransmissions: Tradeoff in configuring nb-iot random access channels," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3796–3805, 2019.

[27] J. Hoydis, F. A. Aoudia, S. Cammerer, M. Nimier-David, N. Binder, G. Marcus, and A. Keller, "Sionna rt: Differentiable ray tracing for radio propagation modeling," in *2023 IEEE Globecom Workshops (GC Wkshps)*, pp. 317–321, 2023.

[28] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, "An open-source lora physical layer prototype on gnu radio," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020.

[29] Z. Xu, S. Tong, P. Xie, and J. Wang, "From demodulation to decoding: Toward complete lora phy understanding and implementation," *ACM Trans. Sen. Netw.*, vol. 18, jan 2023.

[30] B. Liu, C. Gu, S. He, and J. Chen, "Lophy: A resilient and fast covert channel over lora phy," in *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, IPSN '23, (New York, NY, USA), p. 1–13, Association for Computing Machinery, 2023.

[31] X. Guo, L. Shangguan, Y. He, J. Zhang, H. Jiang, A. A. Siddiqi, and Y. Liu, "Efficient ambient LoRa backscatter with ON-OFF keying modulation," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 641–654, 2022.

[32] X. Guo, L. Shangguan, Y. He, N. Jing, J. Zhang, H. Jiang, and Y. Liu, "Saiyan: Design and implementation of a low-power demodulator for LoRa backscatter systems," in *Proceedings of USENIX NSDI*, 2022.

[33] X. Guo, Y. He, N. Jing, J. Zhang, Y. Liu, and L. Shangguan, "A low-power demodulator for LoRa backscatter systems with frequency-amplitude transformation," *IEEE/ACM Transactions on Networking*, vol. 32, no. 4, pp. 3515–3527, 2024.

[34] F. RADU, A. TIMOFTE, A. BALAN, and F. SANDU, "Lte communications using an sdr platform," in *2020 13th International Conference on Communications (COMM)*, pp. 393–396, 2020.

[35] C.-Y. Ho, R.-G. Cheng, J.-W. Chen, and C.-S. Liu, "Open nb-iot network in a pc," in *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2019.

[36] C. Yu, L. Yu, Y. Wu, Y. He, and Q. Lu, "Uplink scheduling and link adaptation for narrowband internet of things systems," *IEEE Access*, vol. 5, pp. 1724–1734, 2017.

[37] D. Yang, X. Zhang, X. Huang, L. Shen, J. Huang, X. Chang, and G. Xing, "Understanding power consumption of nb-iot in the wild: tool and large-scale measurement," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[38] A. K. Sultania, C. Delgado, and J. Famaey, "Implementation of nb-iot power saving schemes in ns-3," in *Proceedings of the 2019 Workshop on Next-Generation Wireless with Ns-3*, WNGW 2019, (New York, NY, USA), p. 5–8, Association for Computing Machinery, 2019.