LPWAN in the TV White Spaces: A Practical Implementation and Deployment Experiences

MAHBUBUR RAHMAN, City University of New York, USA
DALI ISMAIL, VENKATA P. MODEKURTHY, and ABUSAYEED SAIFULLAH, Wayne State University, USA

Low-Power Wide-Area Network (LPWAN) is an enabling Internet-of-Things technology that supports long-range, low-power, and low-cost connectivity to numerous devices. To avoid the crowd in the limited ISM band (where most LPWANs operate) and cost of licensed band, the recently proposed Sensor Network over White Spaces (SNOW) is a promising LPWAN platform that operates over the TV white spaces. As it is a very recent technology and is still in its infancy, the current SNOW implementation uses the Universal Software Radio Peripheral devices as LPWAN nodes, which has high costs (≈\$750 USD per device) and large form-factors, hindering its applicability in practical deployment. In this article, we implement SNOW using low-cost, low form-factor, low-power, and widely available commercial off-the-shelf (COTS) devices to enable its practical and large-scale deployment. Our choice of the COTS device (TI CC13x0: CC1310 or CC1350) consequently brings down the cost and form-factor of a SNOW node by 25× and 10×, respectively. Such implementation of SNOW on the CC13x0 devices, however, faces a number of challenges to enable link reliability and communication range. Our implementation addresses these challenges by handling peak-to-average power ratio problem, channel state information estimation, carrier frequency offset estimation, and near-far power problem. Our deployment in the city of Detroit, Michigan, demonstrates that CC13x0-based SNOW can achieve uplink and downlink throughputs of 11.2 and 4.8 kbps per node, respectively, over a distance of 1 km. Also, the overall throughput in the uplink increases linearly with the increase in the number of SNOW nodes.

CCS Concepts: • **Networks** → **Network architectures**; **Network performance evaluation**; *Network mobility*; Network protocol design; • **Computer systems organization** → *Sensor networks*;

Additional Key Words and Phrases: LPWAN, SNOW, white spaces, OFDM

ACM Reference format:
Mahbubur Rahman, Dali Ismail, Venkata P. Modekurthy, and Abusayeed Saifullah. 2021. LPWAN in the TV White Spaces: A Practical Implementation and Deployment Experiences. *ACM Trans. Embed. Comput. Syst.* 20, 4, Article 30 (May 2021), 26 pages.
https://doi.org/10.1145/3447877

This work was supported by NSF through grants CAREER-1846126 and CNS-2006467.

Authors' addresses: M. Rahman, City University of New York, Queens College Science Building, 65-30 Kissena Boulevard, Flushing, NY 11367, USA; email: mdmahbubur.rahman@qc.cuny.edu; D. Ismail, V. P. Modekurthy, and A. Saifullah, Wayne State University, 5057 Woodward Avenue, Suite 3010, Detroit, MI 48202, USA; emails: {dali.ismail, modekurthy, saifullah}@wayne.edu.

1539-9087/2021/05-ART30 \$15.00
https://doi.org/10.1145/3447877

1 INTRODUCTION

Low-Power Wide-Area Network (LPWAN) is an emerging communication technology that supports long-range, low-power, and low-cost connectivity to numerous devices. It is regarded as a key technology to drive the **Internet-of-Things (IoT)**. Due to their escalating demand, recently multiple LPWAN technologies have been developed that operate in the licensed/cellular (NB-IoT, LTE-M, 5G, etc.) or unlicensed/non-cellular (**Sensor Network over White Spaces (SNOW)**, LoRa, SigFox, etc.) bands [21]. Most of the non-cellular technologies operate in the sub-1-GHz ISM band except SNOW and WEIGHTLESS-W that operate in the TV white spaces [21, 43].

White spaces are the allocated but locally unused TV spectrum (54–698 MHz in the US) that can be used by unlicensed devices as the secondary users. Compared to the crowded ISM band, white spaces offer less crowded and much wider spectrum in both urban and rural areas, boasting an abundance in rural and suburbs [50]. Due to their low frequency, white spaces have excellent propagation and obstacle penetration characteristics enabling long-range communication. Thus, they hold the potentials for LPWAN to support various IoT applications. To our knowledge, WEIGHTLESS-W (which, to the best of our knowledge, has been decommissioned [21]) and SNOW [51] are the only two efforts to exploit the TV white spaces for LPWAN. Initially introduced in Reference [49], SNOW is a highly scalable LPWAN technology offering reliable, bi-directional, concurrent, and asynchronous communication between a **base station (BS)** and numerous nodes.

Despite its promise as a LPWAN technology, SNOW has not yet received sufficient attention from the research community due to its limited availability for practical deployment. The SNOW implementation in Reference [51], which is also available as open-source [4], uses **Universal Software Radio Peripheral (USRP)** devices as LPWAN nodes, hindering the applicability of this technology in practical and large-scale deployment. USRP is a hardware platform developed for software-defined radio applications [6]. Using them as the SNOW node limits the practical deployment of SNOW in real-world applications due to several factors including its high cost and large form-factor. Currently, a USRP B200 device with a half-duplex radio costs ≈$750 USD. As such, it inherently becomes costly to deploy a large-scale SNOW network. Today, IoT applications including smart city (e.g., waste management and smart grid), transportation and logistics (e.g., connected vehicles), agricultural and smart farming (i.e., Microsoft FarmBeats), process management (e.g., oil field monitoring), and healthcare require collection of information from thousands of IoT nodes [21].

In this article, we address the above practical limitations of the existing SNOW technology by implementing it on low-cost and low form-factor **commercial off-the-shelf (COTS)** devices. Through this implementation, we demonstrate that any COTS device that has a **programmable physical layer (PHY)**, operates in the white spaces, and supports **amplitude-shift-keying (ASK)** or **binary phase-shift-keying (BPSK)** may work as a SNOW node in practical deployments. Along with our original TI CC1310-based SNOW implementation in Reference [40], we thus implement SNOW on TI CC1350 that has a programmable PHY, costs approximately $30 USD (retail price) and is 10× smaller than a USRP B200 device, thereby making SNOW adoptable in practical IoT applications. Additionally, we emphasize the fact that this implementation maybe is adaptable in the diverse range of COTS IoT devices with minimal efforts (e.g., only needs to deal with new development software), which makes it highly portable and practical choice for SNOW.

The existing USRP-based SNOW implementation does not face the following practical challenges due to the expensive and powerful hardware design of USRP (as reflected by evaluation in References [49–51]), which the implementation on CC13x0 (CC1310 or CC1350) has to address. First, due to its **orthogonal frequency division multiplexing–(OFDM)** based design, the SNOW BS transmitter is subject to high **peak-to-average power ratio (PAPR)**. Thus, the overall reliability at the CC13x0 device during downlink communication degrades severely. Second, due to the asymmetric bandwidth requirements of the SNOW BS and the nodes, **channel state information (CSI)** estimation between the BS and a CC13x0 device plays a vital role in both uplink and downlink communications. Without CSI estimation, the overall reliability and the communication range decreases. Third, **Carrier frequency offset (CFO)** needs to be handled robustly as its effects are much more pronounced in the low-cost CC13x0 devices, leading to severe **inter-carrier-interference (ICI)**. ICI decreases the overall bitrate in both uplink and downlink communications. While these challenges are quite common in the wireless domain, due to the novel design of SNOW, the existing solutions/approaches may not be adopted in SNOW. Along with addressing these challenges, through this new implementation, we also make SNOW resilient to the classic near-far power problem. Due to the near-far power problem, where a far node's transmission gets buried under a near node's transmission radiation, the reliability in the uplink communication may be degraded. We thus address the above challenge as well. Specifically, we make the following key contributions.

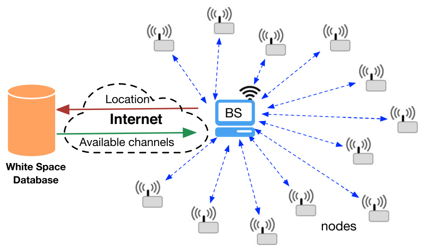
- We implement SNOW for practical deployment on CC13x0s to work as SNOW nodes. Compared to the current USRP-based SNOW implementation, the cost and the form-factor of a single CC13x0-based SNOW node are decreased approximately 25× and 10×, respectively.
- In our implementation, we address several practical challenges including the PAPR problem, CSI and CFO estimation, and near-far power problem. Specifically, we propose a data-aided CSI estimation technique that allows a CC13x0 device to communicate directly with the SNOW BS from a distance of approximately 1 km. Additionally, we propose a pilot-based CFO estimation technique that takes into account the device mobility and increases reliability in both uplink and downlink communications. Finally, we address the near-far power problem in SNOW through an **adaptive transmission power control (ATPC)** protocol that improves the reliability in the uplink communications.
- We experiment with the CC13x0-based SNOW implementation through deployment in the city of Detroit, Michigan. Our results demonstrate that we achieve an uplink throughput of 11.2 kbps per node. Additionally, our overall uplink throughput increases *linearly* with the increase in the number of SNOW nodes. In the downlink, we achieve a throughput of 4.8 kbps per node. Compared to a typical LoRa deployment (channel bandwidth: 500 kHz, spreading factor: 7, and coding rate: 4/5), our uplink throughput is approximately 3.7× higher when 5 nodes transmit to a gateway that can receive concurrent packets using 3 channels.

In the rest of the article, Section 2 provides an overview of SNOW and TI CC13x0. Section 3 presents our SNOW implementation. Section 4 describes the near-far power problem in SNOW. Sections 5 and 6 analyze the deployment cost and performance of our CC13x0-based SNOW, respectively. Section 7 overviews related work. Finally, Section 8 concludes our article.

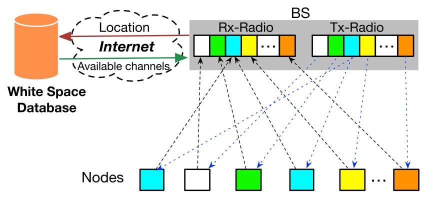
2 BACKGROUND AND SYSTEM MODEL

2.1 An Overview of SNOW

In this section, we provide a brief overview of SNOW. Its complete design and description is available in Reference [39]. SNOW is a highly scalable LPWAN technology operating in the TV white spaces. It supports asynchronous, reliable, bi-directional, and concurrent communication between a BS and numerous nodes. Due to its long-range, SNOW forms a star topology allowing the BS and the nodes to communicate directly, as shown in Figure 1(a). The BS is powerful, Internet-connected, and line-powered while the nodes are power-constrained and do not have access to



(a) Network Architecture



(b) Dual-radio BS and subcarriers

Fig. 1. The SNOW architecture [50].

the Internet. To determine white space availability in a particular area, the BS queries a cloud-hosted geo-location database via the Internet. A node depends on the BS to learn its white space availability. In SNOW, all the complexities are offloaded to the BS to make the node design simple. Each node is equipped with a single half-duplex radio. To support simultaneous uplink and downlink communications, the BS uses a dual-radio architecture for **reception (Rx)** and **transmission (Tx)**, as shown in Figure 1(b).

The SNOW PHY uses a distributed implementation of OFDM called *D-OFDM*. D-OFDM enables the BS to receive concurrent transmissions from *asynchronous* nodes using a single-antenna radio (Rx-radio). Also, using a single-antenna radio (Tx-Radio), the BS can transmit different data to different nodes concurrently [41, 42, 44, 49–51]. Note that the SNOW PHY is different from MIMO radio design adopted in other wireless domains such as LTE, WiMAX, and 802.11n [50] as the latter use multiple antennas to enable multiple transmissions and receptions. The BS operates on a wideband channel split into orthogonal narrowband channels/subcarriers (Figure 1(b)). Each node is assigned a single subcarrier. For encoding and decoding, the BS runs **inverse fast Fourier transform (IFFT)** and **global fast Fourier transform (G-FFT)** over the entire wideband channel, respectively. When the number of nodes is no greater than the number of subcarriers, every node is assigned a unique subcarrier. Otherwise, a subcarrier is shared by more than one node. SNOW supports ASK and BPSK modulation techniques, supporting different bitrates.

The nodes in SNOW use a lightweight CSMA/CA (carrier sense multiple access with collision avoidance)-based **media access control (MAC)** protocol similar to TinyOS [5]. The nodes can autonomously transmit, remain in receive mode, or sleep. Specifically, when a node has data to send, it wakes up by turning its radio on. Then it performs a random back-off in a fixed initial back-off window. When the back-off timer expires, it runs Clear Channel Assessment. If the subcarrier is clear, then it transmits the data. If the subcarrier is occupied, then the node makes a random back-off in a fixed congestion back-off window. After this back-off expires, if the subcarrier is clean the node transmits immediately. This process is repeated until it makes the transmission and gets an **acknowledgment (ACK)**.

2.2 An Overview of TI CC13x0 LaunchPads

Texas Instruments introduced TI CC1310 and TI CC1350 LaunchPads as a part of the SimpleLink **microcontroller (MCU)** platform to support ultra-low-power and long-range communication having a Cortex-M3 processor with 8-KB SRAM and up to 128 KB of in-system programmable storage [2, 40]. With a small form-factor (length: 8 cm, width: 4 cm), both CC1310 and CC1350 are designed to operate in the lower frequency bands (287–351 MHz, 359–527 MHz, and 718–1054 MHz) including the TV band. As an added feature, CC1350 can also operate in the 2.4-GHz band while using Bluetooth low energy radio. The packet structure of the CC13x0 devices includes

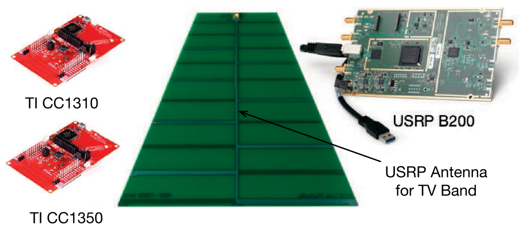


Fig. 2. Devices used in our SNOW implementation. A node is a CC1310 or CC1350 device. The BS has two USRP B200s, each having its own antenna. An antenna is approximately 2× bigger than a B200.

a `preamble`, followed by `sync word`, `payload length`, `payload`, and `CRC`, chronologically. They support different data modulation techniques including **Frequency Shift Keying (FSK)**, Gaussian FSK, **On-Off Keying (OOK)**, and a proprietary long-range modulation. They are capable of using a Tx/Rx bandwidth that ranges between 39 and 3,767 kHz. Additionally, with a supply voltage in the range of 1.8 to 3.8 V, their Rx and Tx current consumption is 5.4 mA and 13.4 mA at +10 dBm, respectively, offering ultra-low-power communication. The greatest advantage is that they have a programmable and reconfigurable physical layer, offering flexibility and feasibility for customized protocol implementation.

3 SNOW IMPLEMENTATION ON TI CC13X0

The original SNOW implementation in Reference [51] uses the USRP hardware platform for both the BS and the nodes. In our implementation, we use the CC13x0 devices as SNOW nodes and USRP in the BS (Figure 2). For BS implementation, we adopt the open-source code provided in Reference [4]. The BS uses two half-duplex USRP devices (Rx-Radio and Tx-Radio), each having its own antenna. Also, the BS is implemented on the GNURadio software platform that gives a high magnitude of freedom to perform baseband signal processing [3]. In the following, we explore a number of implementation considerations and feasibility for a CC13x0 device to work as a SNOW node in practical deployments. First, we show how to configure a CC13x0 device to make it work as a SNOW node. We then address the practical challenges (e.g., PAPR problem, CSI estimation, and CFO estimation) associated with our CC13x0-based SNOW implementation.

3.1 Configuring TI CC13x0

We configure the subcarrier center frequency, bandwidth, modulation, and the Tx power by setting appropriate values to the CC13x0 command inputs `centerFreq, rxBw, modulation`, and `txPower`, respectively, using **Code Composer Studio (CCS)** provided by Texas Instruments [40]. A graphical user interface alternative to CCS is *SmartRF Studio*. The MAC protocol of SNOW in CC13x0 is implemented on top of the example CSMA/CA project that comes with CCS. Note that the functionalities of a SNOW node are very simple and may be incorporated easily in the IoT devices that have both storage and computational limitations like the CC13x0 devices.

3.2 Peak-to-Average Power Ratio Observation

By transmitting on a large number of subcarriers simultaneously (in the downlink), the BS suffers from a traditional OFDM problem called PAPR. PAPR of an OFDM signal is defined as the ratio of the maximum instantaneous power to its average power. In the SNOW downlink communications (i.e., BS to nodes), after the IFFT is performed by the BS, the composite signal can be represented as $x(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{j2\pi f_k t}, 0 \le t \le NT$. Here, X_k is the modulated data symbol for node $k = \{0, 1, \ldots, N-1\}$ on subcarrier center frequency $f_k = k\Delta f$, where $\Delta f = \frac{1}{NT}$ and T is the symbol

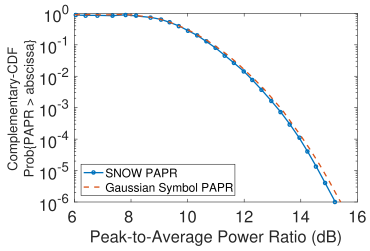


Fig. 3. PAPR distribution of D-OFDM signal in Tx-Radio.

period. Therefore, the PAPR may be calculated as

$$\mathrm{PAPR}[x(t)] = 10\log_{10}\left(\frac{\max\limits_{0\le t\le NT}[|x(t)|^2]}{P_{\mathrm{avg}}}\right) dB.$$

Here, the average power $P_{\mathrm{avg}} = E[|x(t)|^2]$. A node's signal detection on its subcarrier is very sensitive to the nonlinear signal processing components used in the BS, i.e., the digital-to-analog converter and **high power amplifier (HPA)**, which may severely impair the **bit error rate (BER)** in the nodes due to the induced spectral regrowth. If the HPA does not operate in the linear region with a large power back-off due to high PAPR, then the out-of-band power will exceed the specified limit and introduce severe ICI [24]. Moreover, the in-band distortion (constellation tilting and scattering) due to high PAPR may cause severe performance degradation [25]. It has been shown that the PAPR reduction results in significant power saving at the transmitters [10].

As shown in Figure 3, the PAPR in the SNOW downlink communications (for $N = 64$) follows the Gaussian distribution. Thus, the peak signal occurs quite rarely and the transmitted D-OFDM signal will cause the HPA to operate in the nonlinear region, resulting in a very inefficient amplification. To illustrate the power efficiency of the HPA for $N = 64$, let us assume the probability of the clipped D-OFDM frames is less than 0.01%. We thus need to apply an input back-off [10] equivalent to the PAPR at a probability of 10^{-4}. Here, PAPR $\approx$ 14 dB or 25.12. Thus, the efficiency ($\eta = 0.5/\mathrm{PAPR}$) of the HPA [24] is $\eta = 0.5/25.12 \approx 1.99\%$. Such low efficiency at the HPA motivates us to explore the high PAPR in SNOW for practical deployments. Several uplink PAPR reduction techniques for single-user OFDM systems have been proposed (see survey [24]). However, the characteristics of the downlink PAPR in SNOW, where different data are concurrently transmitted to different nodes, are entirely different from the PAPR observed in a single-user OFDM system. To adopt an uplink PAPR reduction technique used in the single-user OFDM systems for the downlink PAPR reduction in SNOW, each node has to process the entire data frame transmitted by the BS and then demodulate its own data. However, a SNOW node has less computational power and does not apply FFT to decode its data [51], or any other node's data. Thus, the existing PAPR reduction techniques will not work in our implementation.

We propose to handle the PAPR problem in SNOW by using only one subcarrier (called *downlink subcarrier*) for downlink communication. All the nodes use this subcarrier to receive from the BS. Namely, the Tx-Radio transmits only on one subcarrier that is not used by any node for uplink communication. The BS may send any broadcast message, ACK, or data to the nodes using that downlink subcarrier. A node has to switch to the downlink subcarrier to listen to any broadcast message, ACK, or data. The BS may reserve multiple subcarriers as *backup subcarriers* for downlink communication. If the currently used downlink subcarrier becomes overly noisy or unreliable, then it can be replaced by a backup subcarrier. Note that the dual-radio in the BS allows it to receive concurrent packets from a set of nodes (uplink) and transmit broadcast/ACK/data packets to another set of nodes (downlink), simultaneously. The BS can acknowledge several nodes using a single transmission by using a bit-vector of size equals to the number of subcarriers. If the BS receives a packet from a node operating on subcarrier i, then it will set the ith bit in the bit-vector. Upon receiving the bit-vector, that node may get an ACK by looking at the ith bit of the vector. Because of the bit-vector, the downlink ACKs also scale up like the uplink traffic. In the case of different packets for different nodes, the volume of downlink traffic (compared to the uplink traffic) is also practical, since the IoT applications may not require high volume downlink traffic [43].

When a node u transmits to the BS, if another node v sharing the same subcarrier wants to transmit, v senses the channel as busy and refrain from transmitting. When the BS transmits ACK to u on the downlink subcarrier using the Tx-Radio, node v may also transmit to the BS. Since the Tx-Radio at that time is making a downlink transmission, it may not send the ACK upon v's transmission immediately. However, the Tx-Radio can send v's ACK immediately after completing its current downlink transmission. Thus, v may need to wait for ACK for a little longer than the time needed to send a downlink transmission from the BS. A node may go to sleep mode or its next state right after receiving an ACK. However, a node that has not received the ACK for a sent packet should wait for a little longer (e.g., up to one or two downlink transmission times). Note that a very few nodes (sharing the same subcarrier) may be involved in this scenario, since the ACK generation time at the BS is very small. For the same reason, the waiting time for ACK will also not be very long (e.g., up to one or two downlink transmission time). Note that this scenario is quite rare and most of the times the nodes will receive ACK immediately upon transmission.

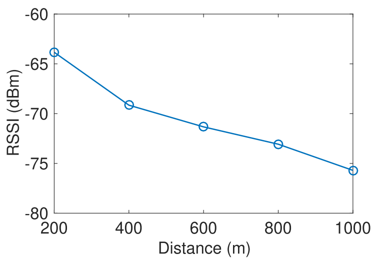
3.3 Channel State Information Estimation

Multi-user OFDM communication requires channel estimation and tracking to ensure high data rate at the BS. One way to avoid channel estimation is to use the **differential phase-shift keying (DPSK)** modulation. DPSK, however, results in a lower bitrate at the BS due to a 3 dB loss in the **signal-to-noise ratio (SNR)** [12]. Additionally, the current SNOW design does not support DPSK modulation. SNR at the BS for each node is different in SNOW. Also, SNR of each node is affected differently due to channel conditions, deteriorating the overall bitrate in the uplinks. Thus, it requires handling of the channel estimation in SNOW.

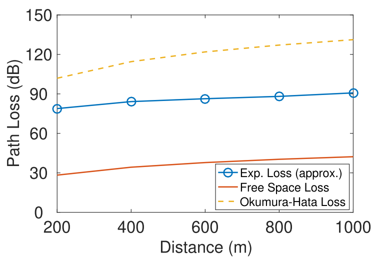
Figure 4 shows the experimentally found **received signal strength indicator (RSSI)**, path loss, and BER at the SNOW BS for a CC1310 device that transmits successive 1000 30-byte (payload) packets from 200- to 1000-m distances, respectively, with a Tx power of 15 dBm, subcarrier center frequency at 500 MHz, and a bandwidth of 98 kHz. Figure 4(a) indicates that the RSSI decreases rapidly with the increase in distance. Also, the path loss in Figure 4(b) shows that it is significantly higher than the theoretical free space loss [45]. We also compare with the Okumura–Hata [45] loss to check if it fits the model; however, it does not. Finally, Figure 4(c) confirms that the BER goes above 10^{-3} (which is not acceptable [22]) beyond 400 m due to the unknown channel conditions. Figure 4(c) also shows that the BER worsens for an increase in the subcarrier bandwidth. Thus, to make our implementation more resilient, we need to incorporate the CSI estimation in SNOW.

We calculate the CSI for each SNOW node independently on its subcarrier. We consider a slow flat-fading model [56], where the channel conditions vary slowly with respect to a single node to BS packet duration. Note that joint-CSI estimation [23, 46] in SNOW is not our design goal, since it would require SNOW nodes to be strongly time-synchronized. Similarly to IEEE 802.16e, we run CSI estimation independently for each node because of their different fading and noise characteristics. In the following, we explain the CSI estimation technique for one node on its subcarrier for each packet. The BS uses the same technique to estimate CSI for all other nodes.

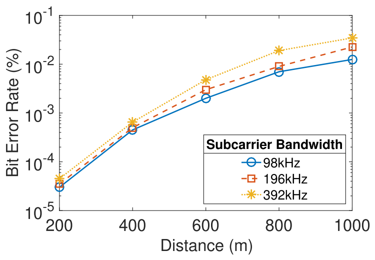
For a node, in a narrowband flat-fading subcarrier, the system is modeled as $y = Hx + w$, where y, x, and w are the receive vector, transmit vector, and noise vector, respectively. H is the



(a) RSSI under varying distance



(b) Path Loss under varying distance



(c) BER under varying distance

Fig. 4. RSSI, path loss, and BER at the SNOW BS for a TI CC1310 node.

channel matrix. We model the noise as additive white Gaussian noise, i.e., a circular symmetric **complex normal (CN)** with $w \sim CN(0, W)$, where the mean is zero and noise covariance matrix W is known. As the subcarrier conditions vary, we estimate the CSI on a short-term basis based on popular approach called *training sequence*. We use the known preamble transmitted at the beginning of each packet. H is estimated using the combined knowledge of the received and the transmitted preambles. To make the estimation robust, we divide the preamble into n equal parts (preamble sequence). For example, n = 4, which is similar to the estimation in IEEE 802.11.

Let the preamble sequence be $(p_1, p_2, \dots, p_n)$, where vector p_i is transmitted as $y_i = Hp_i + w_i$. Combining the received preamble sequences, we get $Y = [y_1, \dots, y_n] = HP + W$, where $P = [p_1, \dots, p_n]$ and $W = [w_1, \dots, w_n]$. With combined knowledge of Y and P, channel matrix H is estimated. Similarly to the CSI estimation in the uplink communications by the BS, each node also estimates the CSI during its downlink communications. Note that the computational complexity of CSI estimation at the nodes is lightweight, since each SNOW packet has a 32-bit preamble [51], divided into four equal parts. A node thus processes a vector of only 8 bits at a time.

3.4 Carrier Frequency Offset Estimation

Multi-user OFDM systems are very sensitive to the CFO between the transmitters and the receiver. CFO causes the OFDM systems to lose orthogonality between subcarriers, which results in severe ICI. A transmitter and a receiver observe CFO due to (i) the mismatch in their local oscillator frequency as a result of hardware imperfections and (ii) the relative motion that causes a Doppler shift. ICI degrades the SNR between an OFDM transmitter and a receiver, which results in significant BER. Thus, we investigate the needs for CFO estimation in our implementation. The loss in SNR due to the CFO between the SNOW BS and a node can be estimated as $SNR_{loss} = 1 + \frac{1}{3}(\pi\delta f T)^2 \frac{E_s}{N_0}$ [38], where δf is the frequency offset, T is the symbol duration, E_s is the average received subcarrier energy, and $N_0/2$ is the two-sided spectral density of the noise power.

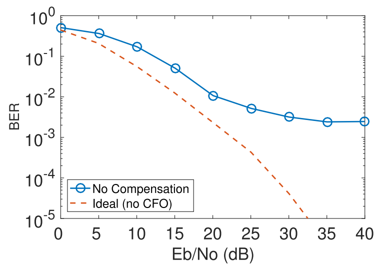


Fig. 5. BER at different E_b/N_0.

To observe the effects of CFO, we choose two neighboring orthogonal subcarriers in the BS and send concurrent packets from two nodes at 200m distance. Each node sends successive one thousand 30-byte packets. We repeat this experiment varying the transmission powers at the nodes to generate signals with different E_b/N_0, where E_b is the average energy per bit in the received signals. Figure 5 shows the BER at the BS while receiving packets from these two nodes. This figure shows that BER is nearly 10^{-3} even for very high E_b/N_0 (≈40 dB), which is also very high compared to the theoretical BER [14]. Thus, CFO is heavily pronounced in SNOW. The distributed and asynchronous nature of SNOW does not allow CFO estimation similar to the traditional multi-user OFDM systems. While the USRP-based SNOW implementation provides a trivial and *coarse* CFO estimation, it is not robust and does not account for the mobility of the nodes [51]. We propose a pilot-based robust CFO estimation technique, combining both coarse and finer estimations, which accounts for the mobility of the nodes as well. We use training symbols for CFO estimation in an ICI free environment for each node independently, while it joins the network by communicating with the BS using a non-overlapping *join subcarrier*.

We explain the CFO estimation technique between a node and the BS (uplink) on a join subcarrier f based on time-domain samples. Note that the BS keeps running the G-FFT on the entire BS spectrum. We thus extract the corresponding time-domain samples of the join subcarrier by applying IFFT during a node join. The join subcarrier does not overlap with other subcarriers; hence it is ICI-free. If f_{node} and f_{BS} are the frequencies at a node and the BS, respectively, then their frequency offset $\delta f = f_{\text{node}} - f_{\text{BS}}$. For transmitted signal $x(t)$ from a node, the received signal $y(t)$ at the BS that experiences a CFO of δf is given by $y(t) = x(t)e^{j2\pi\delta f t}$. Similar to IEEE 802.11a, we estimate δf based on short and long preamble approach. Note that the USRP-based implementation has considered only one preamble to estimate CFO. In our implementation, the BS first divides a n-bit preamble from a node into short and long preambles of lengths $n/4$ and $3n/4$, respectively. Thus for a 32-bit preamble (typically used in SNOW), the lengths of the short and long preambles are 8 and 24, respectively. The short preamble and the long preamble are used for coarse and finer CFO estimation, respectively. Considering δt_s as the short preamble duration and δf_s as the coarse CFO estimation, we have $y(t - \delta t_s) = x(t)e^{j2\pi\delta f_s(t-\delta t_s)}$.

Since $y(t)$ and $y(t - \delta t_s)$ are known at the BS, we have

$$y(t-\delta t_s)y^*(t) = x(t)e^{j2\pi\delta f_s(t-\delta t_s)}x^*(t)e^{-j2\pi\delta f_s t} = |x(t)|^2 e^{j2\pi\delta f_s - \delta t_s}.$$

Taking angle of both sides gives us the following:

$$\sphericalangle y(t-\delta t_s)y^*(t) = \sphericalangle |x(t)|^2 e^{j2\pi\delta f_s - \delta t_s} = -2\pi\delta f_s \delta t_s.$$

By rearranging the above equation, we get

$$\delta f_s = -\frac{\sphericalangle y(t-\delta t_s)y^*(t)}{2\pi\delta t_s}.$$

Now that we have the coarse CFO δf_s, we correct each time domain sample (say, P) received in the long preamble as $P_a = P_a e^{-ja\delta f_s}$, where $a = \{1, 2, \ldots, A\}$ and A is the number of time-domain samples in the long preamble. Taking into account the corrected samples of the long preamble and considering δt_l as the long preamble duration, we estimate the finer CFO as follows:

$$\delta f = -\frac{\sphericalangle y(t - \delta t_l) y^*(t)}{2\pi \delta t_l}. \tag{1}$$

To this extent, considering the join subcarrier f, the **parts per million (ppm)** on the BS's crystal is given by $\text{ppm}_{\text{BS}} = 10^6 \left(\frac{\delta f}{f}\right)$. Thus, the BS calculates δf_i on subcarrier f_i (assigned for node i) as $\delta f_i = \frac{(f_i * \text{ppm}_{\text{BS}})}{10^6}$. The CFO between the Tx-Radio and the Rx-radio can be estimated using a basic SISO CFO estimation technique [60]. Thus, BS also knows the CFO for the downlink.

We now explain the CFO estimation to compensate for the Doppler shift. Note that if the signal bandwidth is sufficiently narrow at a given carrier frequency and mobile velocity, the Doppler shift can be approximated as a common shift across the entire signal bandwidth [55]. Thus, the Doppler shift in the join subcarrier for a node also represents the Doppler shift at its assigned subcarrier, and hence the estimated CFO in Equation (1) is not affected due to the Doppler Shift. For simplicity, we consider that a node's velocity is constant and the change in Doppler shift is negligible during a single packet transmission in SNOW. Considering δf_d as the CFO due to the Doppler shift, v as the velocity of the node, and θ as the angle of the arrived signal at the BS from the node, we have $f_d = f_i \left(\frac{v}{c}\right) \cos(\theta)$ [55], where f_i is the subcarrier center frequency and c is the speed of light. The node itself may consider its motion as circular and approximate $\theta = \frac{\delta s}{r}$, where δs is the amount of anticipated change in position during a packet transmission and r is the *line-of-sight* distance between the node and BS. Thus, CFO compensation due to the Doppler shift is done at the nodes during uplink communications. In the downlink communications, the BS Tx-Radio can also compensate for the node's mobility as the node can report its Doppler shift to the BS during the uplink communications.

In summary, as the nodes asynchronously transmit, estimating joint-CFO of the subcarriers at the BS is very difficult. We thus use a simple feedback approach for proactive CFO correction in the uplink communications. Specifically, δf_i estimated at the BS for subcarrier f_i is given to the node (during joining process at subcarrier f_i). The node may then adjust its transmitted signal based on δf_i and δf_d, calculated as $(\delta f_i + \delta f_d)$, which will align its signal so that the BS does not need to compensate for CFO in the uplink communications. Such feedback-based proactive compensation scheme was studied before for multi-user OFDM and is also used in **global system for mobile communication (GSM)** [11].

4 HANDLING THE NEAR-FAR POWER PROBLEM

Wireless communication is susceptible to the near-far power problem, especially in Code Division Multiple Access [36]. Multi-user D-OFDM system in SNOW may also suffer from this problem. Figure 6 illustrates the near-far power problem in SNOW. Suppose, nodes A and B are operating on two adjacent subcarriers. Node A is closer to the BS compared to node B. When both nodes A and B transmit concurrently to the BS, the received frequency domain signals from node A and B may look as shown on the right of Figure 6. Here, transmission from node B is severely interfered by the strong radiations of node A's transmission. As such, node B's signal may be buried under node A's signal making it difficult for the BS to decode the packet from node B. A typical SNOW deployment may have such scenarios if the nodes operating on adjacent subcarriers use the same transmission power and transmit concurrently at the BS from different distances.

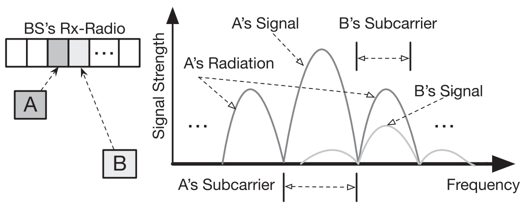
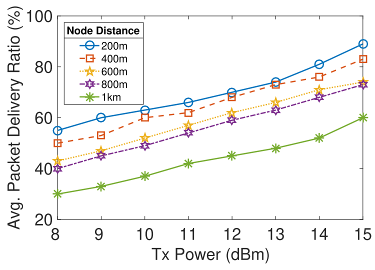
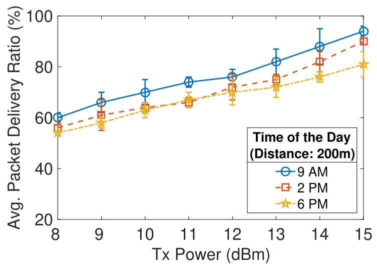


Fig. 6. An illustration of the near-far power problem. B is farther from the BS than A and both transmit concurrently using the same Tx power.



(a) Avg. PDR at different Tx powers



(b) Avg. PDR at different Tx powers and time

Fig. 7. Packet delivery ratio at different Tx powers.

To observe the near-far power problem in SNOW, we run experiments by choosing 3 different adjacent subcarriers, where the middle subcarrier observes the near-far power problem introduced by both subcarriers on its left and right. We place two CC1310 nodes within 20 m of the BS that use the left and the right subcarrier, respectively. We use another CC1310 node that uses the middle subcarrier and is placed at different distances between 200 and 1000 m from the BS. Nodes that are within 20 m of the BS transmit packets continuously with a transmission power of 0 dBm. At each distance, for each transmission power between 8 and 15 dBm, the node that uses the middle subcarrier sends 100 rounds of 1,000 consecutive packets (sends one packet then waits for the ACK and then sends another packet, and so on) to the BS and with a random interval of 0–500 ms. For each transmission power level, at each distance, that node calculates its average **packet delivery ratio (PDR)**. PDR is defined as the ratio of the number of successfully acknowledged packets to the number of total packets sent. We repeat the same experiments for 7 days at 9 AM, 2 PM, and 6 PM.

Figure 7(a) shows that the average PDR increases at each distance with the increase in the transmission power. Figure 7(b) depicts the result for 7-day experiments (only at a distance of 200 m) and shows that the average PDR changes at different time of the day. Overall, Figure 7(a) and (b) confirms that the average PDR increases with the increase in the transmission power. To ensure the energy-efficiency at the nodes, i.e., to find a transmission power that suffices to eliminate the effects of near-far power problem, we propose an adaptive transmission power control for the SNOW design, as described below.

4.1 Adaptive Transmission Power Control

Our design objective for the adaptive Tx power control is to correlate the subcarrier-level Tx power and link quality (i.e., PDR) between each node and the BS. We thus formulate a predictive model to provide each node with a proper Tx power to make a successful transmission to the BS using its assigned subcarrier. Note that our work differs from the work in Reference [32] in fundamental concepts of the network design and architecture. In Reference [32], the authors have considered a multi-hop wireless sensor network based on IEEE 802.15.4 with no concurrency between a set of transmitters and a receiver. Additionally, our model is much more simpler, since we deal with single hop communications. As such, the overheads (i.e., energy consumption and latency at each node) associated with our model are fundamentally lesser than that in Reference [32], or the other techniques developed for multi-hop wireless networks [31, 53]. In the following, we describe our model.

Whenever a node is assigned a new subcarrier or observes a lower PDR, e.g., PDR below **quality of service (QoS)** requirements due to mobility, it runs a lightweight predictive model to determine the convenient Tx power to make successful transmissions to the BS. Our predictive model uses an approximation function to estimate the PDR distribution at different Tx power levels. Over time, that function is modified to adapt to the node's changes. The function is built from the sample pairs of the Tx power levels and PDRs between the node and the BS via a curve-fitting approach. A node collects these samples by sending groups of packets to the BS at different Tx power levels. A node may not be assigned new subcarriers or may not observe lower PDR due to mobility (as per our CSI and CFO estimations) frequently. Thus, the overhead (e.g., energy consumption) for collecting these samples may be negligible compared to the overall network lifetime (which is several years).

Specifically, our predictive model uses two vectors: TP and L, where $TP = \{tp_1, tp_2, \dots, tp_m\}$ contains m different Tx power levels that the node uses to send m groups of packets to the BS and $L = \{l_1, l_2, \dots, l_m\}$ contains the corresponding PDR values at different Tx power levels. Thus, l_i represents the PDR value at Tx power level tp_i. We use the following linear function to correlate between Tx power and PDR,

$$l(tp_i) = a \,.\, tp_i + b. \tag{2}$$

To lessen the computational overhead in the node, we adopt the *least square approximation* technique to determine the unknown coefficients a and b in Equation (2). Thus, we find the minimum of the function $S(a, b)$, where $S(a, b) = \sum |l_i - l(tp_i)|^2$. The minimum of $S(a, b)$ is determined by taking the partial derivatives of $S(a, b)$ with respect to a and b, respectively, and setting them to zero. Thus, $\frac{\partial S}{\partial a} = 0$ and $\frac{\partial S}{\partial b} = 0$ give us

$$a \sum (tp_i)^2 + b \sum tp_i = \sum l_i.tp_i \text{ and}$$
$$a \sum tp_i + b\, m = \sum l_i.$$

Simplifying the above two equations, we find the estimated values of a and b as follows:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \frac{1}{m \sum (tp_i)^2 - (\sum tp_i)^2} \times \begin{bmatrix} m \sum l_i.tp_i - \sum l_i \sum tp_i \\ \sum l_i \sum (tp_i)^2 - \sum l_i.tp_i \sum tp_i \end{bmatrix}$$

Using the estimated values of a and b, the node can calculate the appropriate Tx power as follows:

$$tp = \left[\frac{PDR_{\text{threshold}} - \hat{b}}{\hat{a}} \right] \in TP. \tag{3}$$

Here, $PDR_{\text{threshold}}$ is the threshold set empirically or according to QoS requirements, and [.] denotes the function that rounds the value to the nearest integer in the vector TP.

Now that the initial model has been established in Equation (3), this needs to be updated continuously with the node's changes over time. In Equation (2), both a and b are functions of time that allow the node to use the latest samples to adjust the curve-fitting model dynamically. It is empirically found that (Figure 7(a)) the slope of the curve does not change much over time; hence

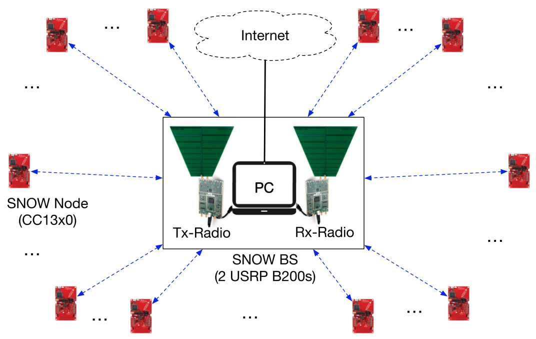


Fig. 8. The SNOW architecture for practical deployment (the PC may be replaced by a Raspberry Pi device; the two USRP B200 devices can be replaced by a USRP B2100 device that has two half-duplex radios.)

a is assumed time-invariant in the predictive model. However, the value of b changes drastically over time (Figure 7(b)). Thus, Equation (2) is rewritten as follows that characterizes the actual relationship between Tx power and PDR,

$$l(tp(t)) = a.tp(t) + b(t).$$

Now, $b(t)$ is determined by the latest Tx power and PDR pairs using the following feedback-based control equation [32],

$$\begin{aligned}\Delta\hat{b}(t) &= \hat{b}(t) - \hat{b}(t+1)\\ &= \frac{\sum_{k=1}^{K}[PDR_{\text{threshold}} - l_k(t-1)]}{K}\\ &= PDR_{\text{threshold}} - l(t-1).\end{aligned} \tag{4}$$

Here, $l(t-1)$ is the average value of K readings denoted as

$$l(t-1) = \frac{\sum_{k=1}^{K} l_k(t-1)}{K}.$$

Here, $l_k(t-1)$, for $k = \{1, 2, \ldots, K\}$, is one reading of PDR during the time period $t-1$ and K is the number of feedback responses at time period $t-1$. Now, the error in Equation (4) is deducted from the previous estimation; hence the new estimation of $b(t)$ can be written as: $\hat{b}(t) = \hat{b}(t-1) - \Delta\hat{b}(t)$. Given the newly estimated $\hat{b}(t)$, the node now can set the Tx power at time t as

$$tp(t) = \left\lceil \frac{PDR_{\text{threshold}} - \hat{b}(t)}{\hat{a}} \right\rceil.$$

5 NETWORK ARCHITECTURE AND DEPLOYMENT COST

In this section, we discuss the practical applicability of our implementation. Figure 8 shows our network view. The SNOW BS is a PC that connects two USRP B200 devices (Tx-Radio and Rx-Radio). The BS is also connected to the Internet. In the BS, a USRP B210 device may be used that has two half-duplex radios. Also, a Raspberry Pi device may be used instead of the PC. All the CC13x0 nodes are battery-powered and directly connected to the BS.

We now analyze the deployment cost of our CC13x0-based SNOW implementation and compare with the original USRP-based SNOW implementation in Reference [51]. Figure 9 shows the total deployment cost of our CC13x0-based SNOW implementation for different numbers of nodes between 1,000 and 10,000. A CC1310 or CC1350 device costs approximately $30 USD (retail price).

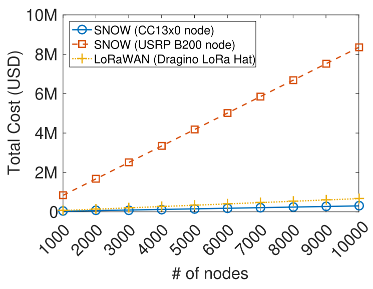


Fig. 9. Practical deployment cost with numerous nodes.

The price for the BS is approximately $1600 USD (two USRP B200 devices $750 USD each, and two antennas $50 USD each). In this comparison, the cost of the PC is not considered, since it is common for both implementations. For SNOW implementation in Reference [51], a node is a USRP B200 device that has an antenna and runs on a Raspberry Pi. A Raspberry Pi device costs approximately $35 USD.

To provide an insight into the deployment cost of a LoRaWAN network, we consider the Dragino LoRa/GPS-Hat (e.g., SX1276 chip) that runs on Raspberry Pi and costs approximately $32 USD (retail price) [15]. We choose this LoRaWAN node, since it has almost identical computational and RF capabilities, compared to the TI CC13x0 devices (e.g., both have the Cortex-M series MCU, similar energy profiles, same set of sensors, and software support). In addition, we consider a LoRaWAN gateway that costs approximately $299 USD and can receive packets on multiple channels simultaneously [1]. We rule out cheaper LoRaWAN devices (costs ≈$10 USD) from the calculation, since they do not have a similar profile as CC13x0 and do not provide software support.

As shown in Figure 9, to deploy an LPWAN with 1000 nodes, the CC13x0-based SNOW implementation may cost approximately $31.6K USD, compared to $836.6K USD for the USRP-based SNOW implementation proposed in Reference [51], and $67.3K USD for the Dragino LoRa-Hat-based LoRaWAN. For a deployment of 10,000 nodes, the costs are $301.6K, $8.3M, and $670.3K USD for CC13x0-based SNOW implementation, USRP-based SNOW implementation, and Dragino LoRa-Hat-based LoRaWAN, respectively. As shown in Figure 9, the cost of each LPWAN increases linearly with the increase in the number of nodes. However, the cost of our CC13x0-based SNOW implementation in unnoticeable. Our new implementation of SNOW on the CC13x0 devices thus becomes highly scalable in terms of cost, making SNOW deployable for practical applications.

6 EVALUATION

In this section, we provide an extensive evaluation (both uplink and downlink performances) of our CC13x0-based SNOW implementation, considering both stationary and mobile CC13x0 nodes.

6.1 Setup

Figure 10 shows our deployment in the city of Detroit, Michigan. Specifically, we deploy 22 CC1310 devices and 3 CC1350 devices (25 CC13x0 devices in total). Due to our limited resources (e.g., batteries/power supply options outdoors), we create five (denoted by black pins) different clusters from 25 nodes at distances 200, 400, 600, 800, and 1,000 m from the BS (denoted by orange pin as well as a label). In each cluster, we randomly place 5 CC1310 devices or a mixture of CC1310 and CC1350 devices that are connected to a laptop via a USB hub so that they can operate without interruptions. In our deployment, one group of nodes may be hidden from another group of nodes, thus creating hidden terminal scenarios. For example, the nodes at 1.05 km and 800 m (see Figure 10) may be hidden from each other, since they are placed in the opposite directions of the

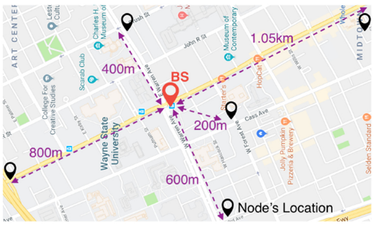


Fig. 10. SNOW deployment in Detroit, Michigan.

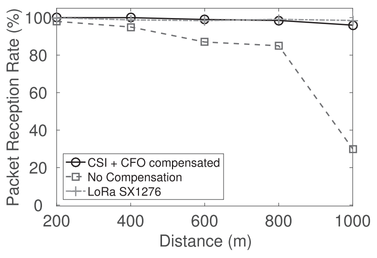
BS. We use a white space channel with frequency band 500–506 MHz and split into 29 (numbered 1–29) overlapping (50%) orthogonal subcarriers, each 400 kHz wide. The USRP-based SNOW also uses a similar subcarrier bandwidth [51]. We use the 28th subcarrier as the join subcarrier and the 26th subcarrier as the downlink subcarrier. We do not use the 29th and the 27th subcarriers so that the join subcarrier may remain ICI-free. The remaining 25 subcarriers are assigned to different nodes.

We use the packet structure of the CC13x0 devices (`preamble`: (32 bits), `sync word`: (32 bits), `paylod length`:, `payload`: variable length, and `CRC` (16 bits)). Our default payload length is 30 bytes and contains random data. Our default bandwidth at the CC13x0 nodes is 39kHz. We use OOK modulation supported by the CC13x0 devices. Unlike the USRP-based SNOW, we do not use any spreading factor. Since the subcarrier bandwidths at the BS and the CC13x0 nodes are 400 and 39 kHz, respectively, the oversampling at each subcarrier in the BS compensates for the spreading factor. Our default transmission power at the BS and the nodes is 15 dBm. However, a CC13x0 device may choose to operate with any transmission power between 0 and 15 dBm, as needed by our ATPC model. The receive sensitivity at the BS is set to −114 dBm, as per the white space regulations [43]. Unless stated otherwise, these are the default parameter settings.

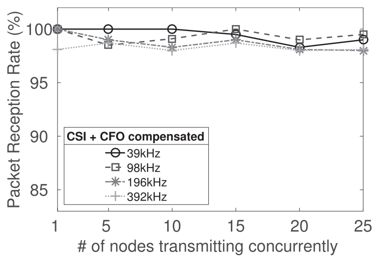
6.2 Reliability over Long Distance

6.2.1 Achievable Distance. We first test the achievable communication range of CC13x0-based SNOW. We take one CC13x0 device and transmit to the BS from different distances between 200 and 1,000 m. We keep our antenna height at 3 m above the ground for both the BS and node. At each distance, each CC13x0 transmits 1,000 packets with a random interval between 0 and 500 ms. The transmission power is set to 15 dBm. To show comparison, we repeat the same experiments without compensating for CSI and CFO as well. Additionally, we test the achievable distance between two LoRa SX1276 devices (bandwidth: 125 kHz, spreading factor: 12, coding rate: 4/5) with the above settings. We choose a bandwidth of 125 kHz in LoRa, since it is the closest value compared to the bandwidth in CC13x0 (39 kHz).

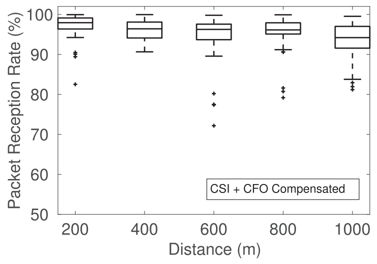
Figure 11(a) shows that the **packet reception rate (PRR)** at the SNOW BS when packets are sent with and without compensating for CSI and CFO, comparing with LoRa. PRR is defined as the ratio of the number of successfully decoded packets to the number of total packets sent. As shown in this figure, with CSI and CFO compensation, the BS achieves 95% of PRR from a distance of 1 km . Without CSI and CFO compensation, the PRR at the BS is as low as 30% from 1 km distance. This figure also shows that a LoRa SX1276 device can deliver packets to another over 1 km with a PRR of 96%, which is similar to the CC13x0-based SNOW node (CSI and CFO compensated). The results thus demonstrate that SNOW on the new platform is highly competitive against LoRa, an LPWAN leader that operates in the ISM band. Additionally, we find that beyond approximately 1 km, PRR stars decreasing in our implementation. Our best guess is that if we can place the BS or



(a) Packet reception rate vs. distances



(b) Packet reception rate in uplink



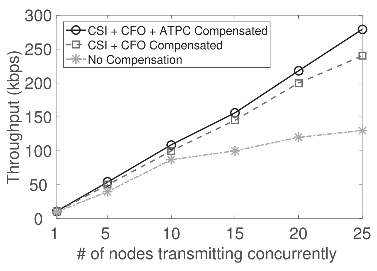
(c) Packet reception rate in downlink

Fig. 11. Reliability in long distance communication.

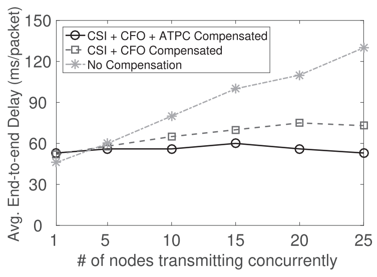
the node at a higher altitude (FCC allows up to 30 m), we may achieve high reliability over much longer communication range.

6.2.2 Uplink Reliability. To show the uplink reliability under concurrent transmissions from different numbers of nodes (CFO and CSI compensated), we transmit from 1, 5, 10, 15, 20, and 25 nodes to the BS, respectively. In this experiment, all the nodes are distributed within 200 and 1,000 m of the BS, as shown in Figure 10. Each node uses different subcarrier bandwidths between 39 and 392 kHz. For each bandwidth starting from 39 kHz, a node sends consecutive 1,000 packets. Between each bandwidth, a node sleeps for 500 ms. Thus, the BS knows the change in the bandwidth. Note that in practical deployment scenarios, a node can let know the BS of its bandwidth during the joining process. In this experiment, we show the performance of a node for different bandwidths. Figure 11(b) shows that we can achieve up to 99% PRR when 25 nodes transmit concurrently using 39 kHz, and up to 98.1% PRR using 392 kHz. Additionally, the number of concurrent transmissions does not affect the trend in PRR for any given bandwidth. Thus, ensuring high uplink reliability of our CC13x0-based implementation over long distances.

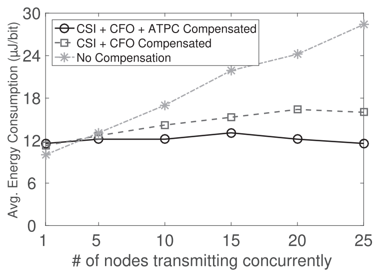
6.2.3 Downlink Reliability. In the downlink, we test the reliability by sending 100 consecutive 30-byte (payload length) packets to each of the 25 nodes that are distributed within 200 and 1,000 m of the BS. We repeat the same experiment 50 times with an interval between 0 and 500 ms. In this experiment, we compensate for both CSI and CFO. Figure 11(c) shows our downlink reliability at different distances observed by different nodes. For better representation, we cluster the nodes that are located approximately at the same distance and plot the PRR against distance. As shown in this figure, the PRR in the downlink is as high as 99% for 75% of the nodes that are approximately 200 m away from the BS. Also, 75% of the nodes that are approximately 1 km away from the BS achieve a PRR of 96%. Thus, this experiment confirms high downlink reliability of our CC13x0-based implementation over long distances.



(a) Throughput



(b) End-to-end delay



(c) Energy consumption

Fig. 12. Network performance in uplink under varying number of nodes.

6.3 Performance in Uplink Communication

In this section, we evaluate the uplink network performance in terms of throughput (achievable bitrate), end-to-end delay (time between the transmission and ACK reception of a packet), and energy consumption. We allow from 1, 5, 10, 15, 20, and 25 nodes to transmit concurrently to the BS, respectively. We distribute the nodes between 200 and 1,000 m in our testbed. Each node transmits 1,000 30-byte (payload length) packets with a random packet interval between 0 and 100 ms. Such packet interval confirms that the node's transmissions are indeed asynchronous to the BS. Each node uses a bandwidth of 39 kHz. We evaluate the uplink network performance for three different cases: (1) nodes or/and BS *compensate* for CSI, CFO, and ATPC; (2) nodes or/and BS *compensate only* for CSI and CFO, but not ATPC; (3) nodes or/and BS *do not compensate* for CSI, CFO, and ATPC. Note that ATPC applies to the nodes only, and hence we use "or/and" in the above three cases. For each case, we run the experiments as long as at least 90% of the packets are delivered to the BS. A node may thus try several retransmissions.

6.3.1 Throughput. Figure 12(a) shows that the BS achieves 279 kbps of throughput when 25 nodes transmit concurrently (case 1), yielding approximately 11.16 kbps per node. Additionally, the overall throughput at the BS increases linearly with an increase in the number of nodes transmitting concurrently. When only CSI and CFO are compensated for, the overall throughput at the BS also increases with an increase in the number of concurrent transmissions, however, it depends on the nodes' distribution (physical) across the network. If there is no near-far power problem, then the overall throughput may be the same as observed in case 1. With no compensation, the throughput per node is approximately 5 kbps, thus more than 2× lesser than case 1. Note that a CC13x0 device can generate a baseband signal with a symbol rate of 11.2 kbaud (OOK modulated). Thus, using a node bandwidth of 39 kHz or 392 kHz will not affect the per node throughput. However, a lower node bandwidth gives higher PRR (Section 6.2) due to longer symbol duration, combating the ICI to some extent. Additionally, if we use any other COTS device that can generate a higher

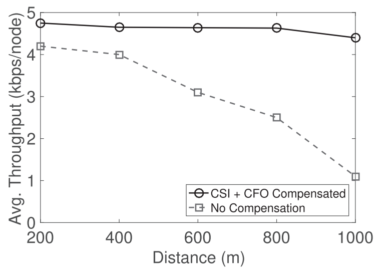


Fig. 13. Average throughput per node in downlink.

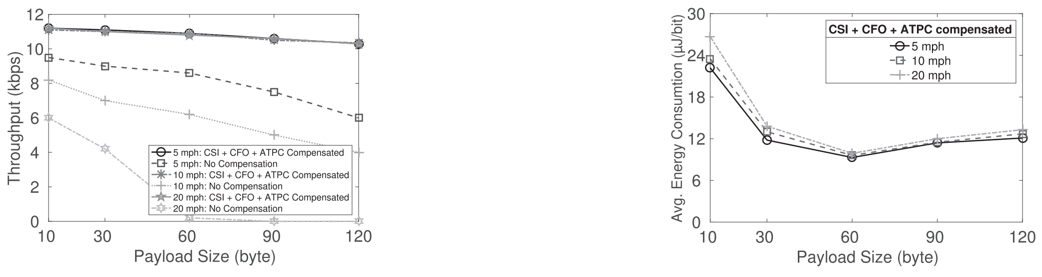
symbol rate for OOK at higher node bandwidth, the per node throughput may also increase with an increase in the node bandwidth. Similarly, adopting a **quadrature amplitude modulation (QAM)** or **frequency modulation (FM)** (both yet to be explored in SNOW) at the nodes may increase the throughput at the SNOW BS. In the future, we shall adopt QAM-based or FM-based modulations in SNOW. Overall, our CC13x0-based SNOW implementation shows high potential for practical deployments of low-rate IoT applications [21, 43].

6.3.2 End-to-end Delay. Figure 12(b) shows the average end-to-end delay per packet at the nodes. When CSI, CFO, and ATPC are compensated for, the average end-to-end delay per packet in the network is 53 ms with 25 concurrent transmissions. Also, for case 1, the average end-to-end delay per packet almost remains constant for any number of concurrent transmissions. For case 2, where only CSI and CFO are compensated for, the average end-to-end delay per packet increases a little bit with an increase in the number of concurrent transmissions. With no compensation, the average end-to-end delay per packet increases almost linearly with an increase in the number of concurrent transmissions. The reason is that a node retransmits several packets several times. Overall, our CC13x0-based SNOW implementation shows great promise for low-latency Industry 4.0 applications and their deployments [35].

6.3.3 Energy Consumption. Figure 12(c) shows the average energy consumption per bit at the nodes. We use the CC13x0 energy profile to calculate the energy consumption during Tx, Rx, and idle time [2]. For case 1, where the CSI, CFO, and ATPC are compensated for, the average energy consumption per bit in the network is approximately 11.6 μJ with 25 concurrent transmissions. Also, the average energy consumption per bit almost remains constant for any number of concurrent transmissions. For case 2, where only CSI and CFO are compensated for, the average energy consumption per bit increases to 16 μJ for 25 concurrent transmissions. Also, when nothing is compensated for, the average energy consumption per packet increases almost linearly with an increase in the number of concurrent transmissions. The reason is that a node retransmits several packets several times. Overall, small energy consumption in case 1 confirms that our CC13x0-based SNOW implementation may host long-lasting deployments.

6.4 Performance in Downlink Communication

In this section, we evaluate the downlink network performance in terms of throughput. The BS sends 1,000 consecutive 30-byte (payload length) packets to each of the 25 nodes. Also, the BS and the nodes compensate for both CSI and CFO. In the downlink, the BS uses a Tx bandwidth of 39 kHz. We repeat the above experiment without compensating for CSI and CFO as well. Figure 13 shows the average throughput per node at different distances. For better representation, we cluster the nodes that are located approximately at the same distance and plot average throughput against the distance. As shown in this figure, a node that is approximately 200 m away from the



(a) Throughput

(b) Energy consumption

Fig. 14. Throughput and energy consumption under mobility.

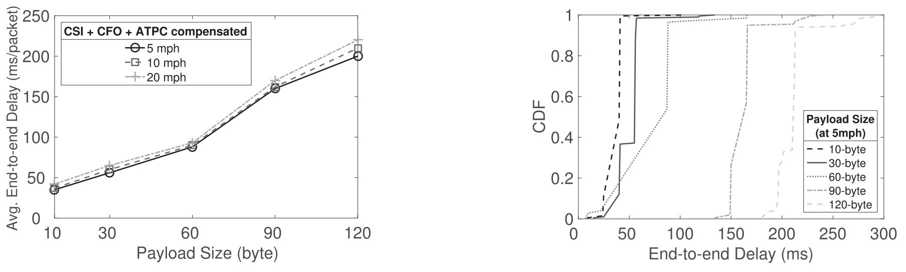
BS can achieve an average downlink throughput of 4.75 kpbs, while both the BS and the node compensate for CSI and CFO. The average throughput remains almost the same as those observed at other distances, up to 1km as well. In contrast, the average throughput drops sharply with an increase in the distance when CSI and CFO are not compensated for. Note that a CC13x0 device can successfully receive an OOK-modulated signal with 4.8 kbaud symbol rate and 39-kHz bandwidth [40]. Overall, our CC13x0-based SNOW implementation holds the potentials for low-rate IoT applications.

6.5 Performance under Mobility

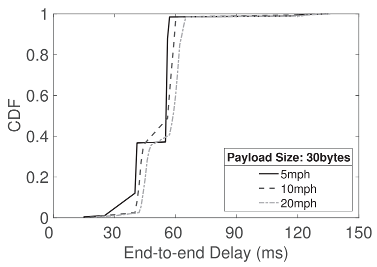
In this section, we evaluate the network performance under CC13x0 node's mobility in terms of throughput, energy consumption, and end-to-end delay. We allow all 25 nodes to transmit concurrently to the BS. However, due to our limited resources, we enable mobility in only one node that is approximately 600 m far from the BS and calculate its performance. All nodes except the mobile node continuously transmit to the BS 30-byte (payload size) packets with a random interval between 0 and 50 ms, using their assigned subcarriers, each 39 kHz wide. Our CFO estimation technique (Section 3.4) allows a mobile node to travel in any direction but at a uniform speed during a packet transmission. We vary the speed of the mobile node for different packets to 5 mph, 10 mph, and 20 mph in any arbitrary direction within our network range. At each speed, we change the payload size of the mobile node between 10 and 120 bytes. For each payload size, the mobile node transmits to the BS 1,000 packets with a random interval between 0 and 50 ms. We run experiments with the above settings for two cases: (1) the mobile node or/and the BS *compensate* for CSI, CFO, and ATPC; (2) the mobile node or/and the BS *do not compensate* for CSI, CFO, and ATPC.

6.5.1 Throughput. Figure 14(a) shows the throughput at the BS (of the mobile node) for different speeds and payload sizes. As this figure suggests, the throughput decreases slightly from 11.18 to 10.3 kbps at 5 mph, 10.35 kbps at 10 mph, and 10.3 kbps at 20 mph for an increase in the payload size between 10 and 120 bytes, as CSI, CFO, and ATPC are compensated for. When the mobile node or/and the BS do not compensate for CSI, CFO, and ATPC, the throughput decreases sharply with an increase in speed and packet size. For example, at 20 mph, the throughput drops to approximately 0 for payload size of 60 bytes. In general, the packet size is susceptible to node's mobility. In fact, if CSI and CFO are not compensated for, then the effects of unknown channel conditions and frequency offset ripple through a longer packet and increase the BER. Thus, our SNOW implementation is resilient and robust under the mobility of the nodes.

6.5.2 Energy Consumption. Figure 14(b) shows an interesting behavior of the energy consumption per bit of the CC13x0 devices. At each speed (CSI, CFO, or/and ATPC compensated), the average energy consumption per bit decreases when we increase the payload size from 10 to 60 bytes.



(a) End-to-end delay (b) CDF of end-to-end delay



(c) CDF of end-to-end delay

Fig. 15. End-to-end delay analysis of mobile node under varying payload length.

But, when we increase the payload size beyond 60 bytes (e.g., 90 and 120bytes), the energy consumption per bit starts to increase. For example, the average energy consumption per bit (at 5mph) for payload sizes 10, 30, 60, 90, and 120 bytes are 22.2, 11.8, 9.3, 11.2, and 12.1 μJ, respectively. At speeds 10 and 20 mph, the trends in energy consumption per bit are also similar to the above trend. To the best of our knowledge, CC13x0 devices show such behavior due to their arbitrary end-to-end delays of packets with different payload lengths (to be discussed in Section 6.5.3). Overall, a payload of length 60 bytes may be preferable (in terms of energy per bit) in the CC13x0-based SNOW implementation.

6.5.3 End-to-end Delay. Figure 15(a) shows that the average per-packet end-to-end delay at the mobile node increases with an increase in speed and payload size. For example, at 5mph, the average per-packet end-to-end delays with payloads of sizes 10, 30, 60, 90, and 120 bytes are 35, 56, 88, 160, 200 ms, respectively; at 20 mph, the average end-to-end delays are 42, 65, 93, 170, 220 ms, respectively. Note that these delays in terms of payload lengths are in fact arbitrary. For example, the ratio between the delays of 30-byte and 10-byte payloads is not $30/10 = 3\times$ but $56/35 = 1.6\times$ at 5 mph and $65/42 \approx 1.5\times$ at 20 mph speed at the node.

Figure 15(b) shows the **cumulative distribution function (CDF)** of the end-to-end delay at a constant speed of 5 mph with varying payload sizes. This figure shows that 60% of the 10-byte (payload length) packets observe an end-to-end delay more than 35 ms, 65% of the 30-byte (payload length) packets observe an end-to-end delay more than 55 ms, 50% of the 60-byte (payload length) packets observe an end-to-end delay more than 90 ms, 98% of the 90-byte (payload length) packets observe an end-to-end delay more than 150 ms, and 95% of the 120-byte (payload length) packets observe an end-to-end delay more than 195 ms. Furthermore, Figure 15(c) shows the CDF of end-to-end delays for a fixed payload length of 30 bytes at varying speed. As this figure shows, 98% of the packets at 5 mph observe an end-to-end delay up to 55 ms, 99.99% of the packets at 10 mph observe an end-to-end delay up to 60 ms, and 98% of the packets at 20 mph observe an end-to-end

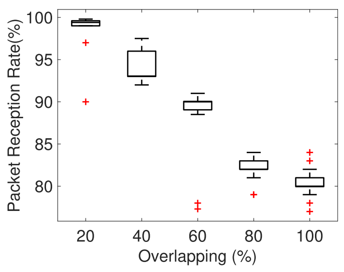


Fig. 16. Performance under interference.

delay up to 65 ms. Overall, Figure 15 confirms that our CC13x0-based SNOW implementation may provide very low latency under the mobility of the nodes.

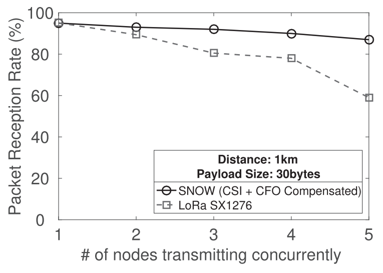
6.6 Performance in the Presence of Interference

In this section, we evaluate the performance of our implementation in the presence of interference. Due to the COVID-19 restrictions, we limit this experiment in an indoor area of (20×30) m^2. We place a USRP B200 device within 10 m of the BS to act as an interferer. Within 200 ms intervals, the interferer randomly operates near the 25 subcarriers used by the CC13x0 devices and transmits random 40-byte payloads. The CC13x0 devices send packets to the BS concurrently and incessantly from distances between 20 and 30 m. We let the interferer overlap 20%, 40%, 60%, 80%, and 100% with the legitimate subcarriers. For each magnitude of overlaps, we run this experiment for 2 minutes. In Figure 16, we show the distribution of PRR at the BS (CSI, CFO, and ATPC are compensated) for 30 runs of the above experiment. As shown in this figure, even with 100% overlap, the PRR at the BS can be as high as 84%. For 60% overlap, the PRR at the BS is as high as 92%. Overall, as we decrease the percentage of overlaps, the PRR increases at the BS. This experiments thus confirms that the impact of external interference is less severe or negligible when the interferer's spectrum partially overlaps with the legitimate subcarriers in our CC13x0-based SNOW implementation.

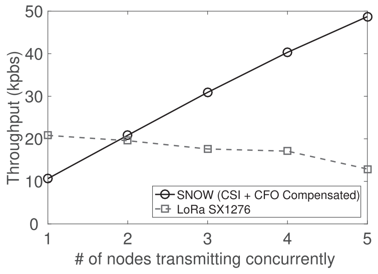
6.7 Performance Comparison with LoRaWAN

In this section, we experimentally compare the performance of our CC13x0-based SNOW implementation with a LoRaWAN network. We have eight Dragino LoRa/GPS-Hat Sx1276 transceivers that can transmit or receive on a single channel. We create a LoRaWAN gateway capable of receiving on three channels simultaneously using three of our LoRa-Hats, while the remaining five devices act as LoRaWAN nodes. For a fair comparison, we allow five SNOW nodes (3 CC1350 devices and 2 CC1310 devices) to transmit to the SNOW BS, allowing only three subcarriers for data Rx/Tx. Similarly, in LoRaWAN, five nodes transmit on three 500-kHz channels using a spreading factor of 7 and a coding rate of $\frac{4}{5}$. In SNOW, the nodes use a subcarrier bandwidth of 392 kHz with no bit spreading factor. While choosing 500 kHz or 392 kHz has no differentiable impact in our CC13x0-based SNOW implementation (as discussed in Section 6.3), we choose the latter due to the configurable Tx bandwidth limitation of the devices. The LoRaWAN gateway uses three adjacent 500-kHz channels in the 915-MHz frequency band (in the US), while the SNOW BS, in this setup, uses three adjacent overlapping subcarriers, numbered 10, 11, and 12 (please refer to Section 6.1 for subcarrier allocation).

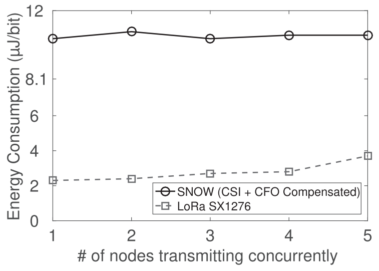
The above configuration for LoRaWAN will result in its best possible throughput and energy-efficiency [51]. Additionally, choosing LoRaWAN's largest spreading factor (e.g., 12) and 125-kHz channel bandwidth for better reliability will make the comparison unfair with SNOW (e.g., SNOW



(a) Packet reception rate at the gateway/BS



(b) Throughput at the gateway/BS



(c) Energy consumption at the nodes

Fig. 17. Uplink performance comparison between SNOW and LoRaWAN.

uses 392-kHz bandwidth and no spreading factor). Each node (for both LoRaWAN and SNOW) transmits one thousand 30-byte (payload size) packets from a distance of approximately 1km to the gateway/BS with a random inter-packet interval between 500 and 1,000 ms and a Tx power of 15 dBm. Each node randomly hops to a different channel/subcarrier after sending 200 packets. In LoRaWAN, the nodes use the pure ALOHA MAC protocol, and thus operating as the Class-A LoRaWAN nodes [21]. In SNOW, the nodes use the lightweight CSMA/CA MAC protocol (as discussed in Section 2.1). In the following, we compare LoRaWAN and SNOW in terms of reliability, throughput, and energy consumption with the above settings. Note that achievable distance comparison between SNOW and LoRaWAN has already been presented in Section 6.2.

6.7.1 Reliability Comparison with Parallel Tx/Rx. Figure 17(a) shows the PRR at the gateway/BS for LoRaWAN and CC13x0-based SNOW implementation under varying number of nodes that transmit concurrently. As shown in this figure, when only one node transmits, the PRR is approximately 95% in both LoRaWAN and SNOW. Also, the PRR of LoRaWAN decreases with the increase in the number of parallel transmissions. For SNOW, it remains almost similar with the increase in the number of parallel transmissions. For example, when 5 nodes transmit in parallel, LoRaWAN achieves a PRR of 59%, compared to 87% in SNOW. Such performance degradation in LoRaWAN happens as it uses an ALOHA-based MAC protocol without any collision avoidance. The PRR of LoRaWAN may increase if we increase the inter-packet interval and will remain the same for SNOW even if we decrease the inter-packet interval.

6.7.2 Throughput Comparison. Figure 17(b) shows the overall throughput (kbps) comparison at the gateway/BS between LoRaWAN and SNOW. As shown in this figure, the throughput at the LoRaWAN gateway is approximately 20.8 kbps, compared to 10.64 kbps at the SNOW BS when only one node transmits. However, the throughput at the SNOW BS surpasses that at the LoRaWAN gateway when 2 or more nodes transmit concurrently. As shown in Figure 17(b), the throughput at the SNOW BS is $\frac{48.12}{12.9} \approx 3.7\times$ higher compared to LoRaWAN when five nodes transmit concurrently. The throughput in LoRaWAN decreases as we increase the number of LoRaWAN nodes because of the following reason. The LoRaWAN nodes adopt the ALOHA-based MAC protocol. Thus, a LoRaWAN node does not check if a channel is free before transmitting a packet, which results in a collision with another ongoing packet transmission (if any) from a different LoRaWAN node in the same channel. Consequently, both packets are lost as well as not considered in the throughput calculation. Packet collisions might happen in our setup, since five LoRaWAN nodes share three channels. Note that our setup is realistic, which emulates the scenario of having hundreds of LoRaWAN nodes under a 64-channel (maximum possible) LoRaWAN gateway. For 100 LoRaWAN nodes, the channel to node ratio is $\frac{64}{100} \approx \frac{3}{5}$, which is also the same in our setup. The SNOW nodes, however, avoid packet collisions on the same channel by adopting the CSMA/CA MAC protocol. Compared to LoRaWAN, our CC13x0-based SNOW implementation thus observes better throughput.

6.7.3 Energy Consumption Comparison. Figure 17(c) shows the per packet energy consumption at the nodes of LoRaWAN and our CC13x0-based SNOW implementation. As the figure shows, when five nodes transmit in parallel, a LoRaWAN node spends 3.7 μJ/bit, compared to 10.6 μJ/bit in SNOW. Here, the per-bit energy consumption in SNOW is slightly higher than that in LoRaWAN. However, this figure shows that the per-bit energy consumption increases in LoRaWAN and remains almost steady in SNOW when the number of concurrent nodes increases. SNOW is designed to enable a large number of concurrent transmissions to the BS and such a tendency in energy consumption shows its energy efficiency under that scenario. However, the number of retransmissions to deliver a packet increases with the increase in the number of nodes in LoRaWAN, thereby increasing the per packet energy consumption. Due to a limited number of devices, we are unable to demonstrate this in real experiment. Note that the LoRaWAN configuration used in this experiment (e.g., channel bandwidth of 500 kHz, spreading factor of 7, coding rate of $\frac{4}{5}$) is the most energy-efficient (for a given Tx power) for an individual LoRaWAN node. As reported in Reference [58], increasing the spreading factor increases energy consumption at the LoRaWAN nodes. Similarly, decreasing the channel bandwidth also increases energy consumption at the LoRaWAN nodes [58].

6.8 Discussion on the Performance of CC1310 and CC1350

As we experiment with both CC1310 and CC1350 devices as the SNOW nodes (Sections 6.2–6.5), we see no noticeable performance difference at the SNOW BS in terms of reliability and throughput, compared to our previous experiments with the CC1310 devices only in Reference [40]. Similarly, there is no noticeable performance difference in terms of end-to-end latency and energy consumption between the two platforms. The reason is that our SNOW implementation is minimally invasive to the IoT devices having almost similar PHY layer properties. We envision that any IoT device with a programmable PHY, capable of operating in the white spaces, and capable of OOK/BPSK modulation may work as a SNOW node by addressing the same set of challenges as described through Sections 3.3–4. In this article, we practically demonstrate this by implementing SNOW on CC1310 and CC1350 devices and showing that their performance is similar.

7 RELATED WORK

Recently, a number of LPWAN technologies have been developed that operate in the licensed (e.g., LTE Cat M1, NB-IoT, EC-GSM-IoT, and 5G) or unlicensed (e.g., LoRa, SigFox, RPMA (INGENU), IQRF, Telensa, DASH7, WEIGHTLESS-N, WEIGHTLESS-P, IEEE 802.11ah, IEEE 802.15.4k, and IEEE 802.15.4g) spectrum [7, 13, 17, 21, 29, 43, 52]. Operating in the licensed band is costly due to high service fee and costly infrastructure. On the contrary, most non-cellular LPWANs including LoRa and SigFox operate in the sub-1-GHz ISM band, for example, between 902 and 928 MHz in the continents of North America and South America. While the ISM band is unlicensed, it is heavily crowded due to the proliferation of LPWANs as well as other wireless technologies in this band. To avoid the high cost of licensed band and the crowd of the ISM band, SNOW was designed to exploit the TV white spaces. As reported in References [8, 21, 43, 50], white spaces are widely available in both urban and rural areas, less crowded (compared to the ISM band), and offer a wider spectrum (compared to the available frequency bands for LPWANs in the ISM band) that a SNOW BS can utilize.

The existing work on white space focused on exploiting the white spaces for broadband access [8, 19, 20, 30, 43, 54, 57, 59, 61–64] and spectrum determination through spectrum sensing [9, 26, 27, 33, 48] and/or geo-location database approach [16, 18, 34, 37, 63]. Alongside, various standards bodies (e.g., IEEE 802.11af, IEEE 802.15.4m, IEEE 802.19.1, IEEE 802.22, IEEE 1900.4a, IEEE 1900.7, and ECMA-392) and industry leaders (e.g., Microsoft and Google) have also targeted the white spaces for unlicensed personal or commercial use [21, 28, 43, 47]. In contrast, SNOW, as a LPWAN, exploits white spaces for highly scalable and wide-area sensor network applications. With the rapid growth of IoT, LPWANs will suffer from crowded spectrum due to long range. It is hence critical to exploit white spaces for IoT. Our article focuses on implementing SNOW using the cheap and widely available COTS devices for practical and scalable deployment.

8 CONCLUSIONS

The recently proposed LPWAN technology—SNOW—has the potential to enable connectivity to numerous IoT devices over long distances. However, the high cost and the large form-factor of the USRP-based SNOW nodes hinder its practical deployments. In this article, we have implemented SNOW for practical deployments using the CC13x0 devices as SNOW nodes. Our CC13x0-based SNOW implementation decreases the cost and the form-factor of a single SNOW node by 25× and 10×, respectively. We have also addressed several practical deployment challenges that include PAPR reduction, CSI estimation, CFO estimation, and near-far power problem. We have deployed our CC13x0-based SNOW in the city of Detroit, Michigan and achieved per node uplink and downlink throughputs of 11.2 and 4.8 kbps, respectively, over a distance of 1 km. Our experiments also show that SNOW can achieve throughput several times higher than LoRaWAN under typical settings. Finally, our extensive experiments have demonstrated the CC13x0-based SNOW as a feasible LPWAN technology that can be deployed practically at low-cost and in large-scale for future IoT applications.

REFERENCES

[1] 2018. Mouser Microchip. Retrieved from https://www.mouser.com/ProductDetail/.
[2] 2019. CC1350 LaunchPad. Retrieved from http://www.ti.com/tool/LAUNCHXL-CC1350.
[3] 2019. GNU Radio. Retrieved from http://gnuradio.org.
[4] 2019. SNOW Base Station. Retrieved from https://github.com/snowlab12/gr-snow.
[5] 2019. TinyOS. Retrieved from http://www.tinyos.net.
[6] 2020. Ettus Research. Retrieved from https://www.ettus.com/product/category/USRP-Bus-Series.
[7] Godfrey Anuga Akpakwu, Bruno J. Silva, Gerhard P. Hancke, and Adnan M. Abu-Mahfouz. 2017. A survey on 5G networks for the Internet of Things: Communication technologies and challenges. *IEEE Access* 6 (2017), 3619–3647.
[8] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. 2009. White space networking with Wi-fi like connectivity. *SIGCOMM Comput. Commun. Rev.* 39, 4 (2009), 27–38.
[9] R. Balamurthi, H. Joshi, C. Nguyen, A. K. Sadek, S. J. Shellhammer, and C. Shen. 2011. A TV white space spectrum sensing prototype. In *DySPAN'11*. IEEE, 297–307.
[10] Robert J. Baxley and G. Tong Zhou. 2004. Power savings analysis of peak-to-average power ratio in OFDM. *IEEE Trans. Cons. Electr.* 50, 3 (2004), 792–798.

[11] J.-J. Beek, P. Borjesson, M. Boucheret, D. Landstrom, J. Arenas, P. Odling, C. Ostberg, M. Wahlqvist, and S. Wilson. 1999. A time and frequency synchronization scheme for multiuser OFDM. *IEEE J. Select. Areas Commun.* 17, 11 (1999), 1900–1914.

[12] J.-J. Beek, O. Edfors, M. Sandell, S. Wilson, and P. Borjesson. 1995. On channel estimation in OFDM systems. In *VTC'95*. 815–819.

[13] Jiming Chen, Kang Hu, Qi Wang, Yuyi Sun, Zhiguo Shi, and Shibo He. 2017. Narrowband internet of things: Implementations and applications. *IEEE IoT J.* 4, 6 (2017), 2309–2314.

[14] Jihoon Choi, Yong-Hoon Lee, Changoo Lee, and Hae Won Jung. 2000. Carrier frequency offset compensation for uplink of OFDM-FDMA systems. In *ICC'00*. IEEE, 425–429.

[15] Dragino. 2018. Dragino LoRa/GPS Hat. Retrieved from http://www.dragino.com/products/lora/item/106-lora-gps-hat.html.

[16] X. Feng, J. Zhang, and Q. Zhang. 2011. Database-assisted multi-AP network on TV white spaces: Architecture, spectrum allocation and AP discovery. In *DySPAN'11*. IEEE, 265–276.

[17] Javier Gozalvez. 2016. New 3GPP standard for IoT. *IEEE Vehic. Technol. Mag.* 11, 1 (2016), 14–20.

[18] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch. 2008. Geo-location database techniques for incumbent protection in the TV white space. In *DySPAN'08*. 1–9.

[19] Kate Harrison, Vidya Muthukumar, and Anant Sahai. 2015. Whitespace evaluation SofTware (WEST) and its applications to whitespace in Canada and Australia. In *DySPAN'15*. IEEE, 47–58.

[20] Shaddi Hasan, Kurtis Heimerl, Kate Harrison, Kashif Ali, Sean Roberts, Anant Sahai, and Eric Brewer. 2014. GSM whitespaces: An opportunity for rural cellular service. In *DYSPAN'14*. 271–282.

[21] Dali Ismail, Mahbubur Rahman, and Abusayeed Saifullah. 2018. Low-power wide-area networks: Opportunities, challenges, and directions. In *Workshop Program of the ICDCN'18*. ACM, 1–8.

[22] Dali Ismail, Mahbubur Rahman, Abusayeed Saifullah, and Sanjay Madria. 2017. RnR: Reverse & replace decoding for collision recovery in wireless sensor networks. In *SECON'17*. IEEE, 1–9.

[23] Ming Jiang, Jos Akhtman, and Lajos Hanzo. 2007. Iterative joint channel estimation and multi-user detection for multiple-antenna aided OFDM systems. *IEEE Trans. Wireless Commun.* 6, 8 (2007).

[24] Tao Jiang and Yiyan Wu. 2008. An overview: Peak-to-average power ratio reduction techniques for OFDM signals. *IEEE Trans. Broadcast.* 54, 2 (2008), 257–268.

[25] Behnam Kamali, Robert Alexander Bennett, and Dyani Camika Cox. 2012. Understanding WiMAX: An IEEE-802.16 standard-based wireless technology. *IEEE Potentials* 31, 5 (2012), 23–27.

[26] H. Kim and K. G. Shin. 2008. Fast discovery of spectrum opportunities in cognitive radio networks. In *DySPAN'08*.

[27] Hyoil Kim and Kang G. Shin. 2008. In-band spectrum sensing in cognitive radio networks: Energy detection or feature detection? In *MobiCom'08*. ACM, 14–25.

[28] Christian Kocks, Alexander Viessmann, Peter Jung, Lei Chen, Qiu Jing, and Rose Qingyang Hu. 2012. On spectrum sensing for TV white space in China. *J. Comput. Netw. Commun.* 2012 (2012), 1–8.

[29] Nikolaos Kouvelas, Vijay S. Rao, R. Venkatesha Prasad, Gauri Tawde, and Koen Langendoen. 2020. p-CARMA: Politely scaling LoRaWAN. In *EWSN*. 25–36.

[30] A. Kumar, A. Karandikar, G. Naik, M. Khaturia, S. Saha, M. Arora, and J. Singh. 2016. Toward enabling broadband for a billion plus population with TV white spaces. *IEEE Commun. Mag.* 54, 7 (2016), 28–34.

[31] L. Li, J. Y Halpern, P. Bahl, Y. Wang, and R. Wattenhofer. 2005. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Trans. Netw.* 13, 1 (2005), 147–159.

[32] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A Stankovic, S. Son, and G. J. Pappas. 2016. ATPC: Adaptive transmission power control for wireless sensor networks. *ACM Trans. Sens. Netw.* 12, 1 (2016), 1–31.

[33] Dongxin Liu, Zhihao Wu, Fan Wu, Yuan Zhang, and Guihai Chen. 2015. FIWEX: Compressive sensing based cost-efficient indoor white space exploration. In *MobiHoc'15*. 17–26.

[34] Y. Luo, L. Gao, and J. Huang. 2015. HySIM: A hybrid spectrum and information market for TV white space networks. In *INFOCOM'15*. IEEE, 900–908.

[35] Venkata Prashant Modekurthy, Dali Ismail, Mahbubur Rahman, and Abusayeed Saifullah. 2018. A utilization-based approach for schedulability analysis in wireless control systems. In *ICII'18*. IEEE, 49–58.

[36] A. Muqattash and M. Krunz. 2003. CDMA-based MAC protocol for wireless ad hoc networks. In *Mobihoc'03*. 1–12.

[37] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl. 2012. SenseLess: A database-driven white spaces network. *IEEE Trans. Mobile Comput.* 11, 2 (2012), 189–203.

[38] Richard van Nee and Ramjee Prasad. 2000. *OFDM for Wireless Multimedia Communications*. Artech House, Inc.

[39] Mahbubur Rahman. 2020. *Low-Power Wide-Area Network Design*. Ph.D. Dissertation. Wayne State University.

[40] Mahbubur Rahman, Dali Ismail, Venkata P. Modekurthy, and Abusayeed Saifullah. 2019. Implementation of LPWAN over white spaces for practical deployment. In *IoTDI'19*. ACM, 178–189.

[41] Mahbubur Rahman, Dali Ismail, and Abusayeed Saifullah. 2018. Demo abstract: Enabling inter-SNOW concurrent P2P communications. In *IoTDI'18*. IEEE, 310–311.

[42] Mahbubur Rahman and Abusayeed Saifullah. 2018. Integrating low-power wide-area networks in white spaces. In *IoTDI'18*. IEEE, 255–260.

[43] Mahbubur Rahman and Abusayeed Saifullah. 2019. A comprehensive survey on networking over TV white spaces. *Perv. Mobile Comput.* 59 (2019), 1–17.

[44] Mahbubur Rahman and Abusayeed Saifullah. 2020. Integrating low-power wide-area networks for enhanced scalability and extended coverage. *IEEE/ACM Trans. Netw.* 28, 1 (2020), 1–14.

[45] Theodore S. Rappaport et al. 1996. *Wireless Communications: Principles and Practice*. Vol. 2. Prentice Hall, Englewood Cliffs, NJ.

[46] C. Ribeiro, M. J. Fernández-Getino Garcia, Víctor P. G. Jiménez, Atílio Gameiro, and A. García Armada. 2008. Uplink channel estimation for multi-user OFDM-based systems. *Wireless Pers. Commun.* 47, 1 (2008), 125–136.

[47] S. Roberts, P. Garnett, and R. Chandra. 2015. Connecting africa using the TV white spaces: From research to real world deployments. In *Proceedings of the 21st IEEE International Workshop on Local and Metropolitan Area Networks*. 1–6.

[48] A. Saeed, K. A. Harras, E. Zegura, and M. Ammar. 2017. Local and low-cost white space detection. In *ICDCS'17*. 503–516.

[49] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. 2016. SNOW: Sensor network over white spaces. In *SenSys'16*. ACM, 272–285.

[50] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, C. Lu, Jie Liu, and R. Chandra. 2017. Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces. In *SenSys'17*. ACM, 1–14.

[51] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Jie Liu, and Ranveer Chandra. 2018. Low-power wide-area network over white spaces. *IEEE/ACM Trans. Netw.* 26, 4 (2018), 1893–1906.

[52] V. Saxena, A. Wallen, T. Tirronen, H. S. Razaghi, J. Bergman, and Y. Blankenship. 2016. On the achievable coverage and uplink capacity of machine-type communications (MTC) in LTE release 13. In *VTC-Fall'16*. 1–6.

[53] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. 2006. Experimental study of concurrent transmission in wireless sensor networks. In *SenSys'06*. ACM, 237–250.

[54] S. Sur and X. Zhang. 2015. Bridging link power asymmetry in mobile whitespace networks. In *INFOCOM'15*. 1176–1184.

[55] S. L. Talbot and B. Farhang. 2007. Mobility and carrier offset modeling in OFDM. In *GLOBECOM'07*. 1–5.

[56] David Tse and Pramod Viswanath. 2005. *Fundamentals of Wireless Communication*. Cambridge University Press.

[57] X. Wang, J. Chen, A. Dutta, and M. Chiang. 2015. Adaptive video streaming over whitespace: SVC for 3-Tiered spectrum sharing. In *INFOCOM'15*. 28–36.

[58] Weitao Xu, Jun Y Kim, Walter Huang, Salil S Kanhere, Sanjay K Jha, and Wen Hu. 2019. Measurement, characterization, and modeling of LoRa technology in multifloor buildings. *IEEE IoT J.* 7, 1 (2019), 298–310.

[59] L. Yang, W. Hou, L. Cao, B Y Zhao, and H. Zheng. 2010. Supporting demanding wireless applications with frequency-agile radios. In *NSDI'10*. 65–80.

[60] Yingwei Yao and Georgios B Giannakis. 2005. Blind carrier frequency offset estimation in SISO, MIMO, and multiuser OFDM systems. *IEEE Trans. Commun.* 53, 1 (2005), 173–183.

[61] Xuhang Ying, Jincheng Zhang, Lichao Yan, Yu Chen, Guanglin Zhang, Minghua Chen, and Ranveer Chandra. 2017. Exploring indoor white spaces in metropolises. *ACM Trans. Intell. Syst. Technol.* 9, 1 (2017), 1–25.

[62] Tan Zhang, Aakanksha Chowdhery, Paramvir (Victor) Bahl, Kyle Jamieson, and Suman Banerjee. 2015. The design and implementation of a wireless video surveillance system. In *MobiCom'15*. ACM, 426–438.

[63] Tan Zhang, Ning Leng, and Suman Banerjee. 2014. A vehicle-based measurement framework for enhancing whitespace spectrum databases. In *MobiCom'14*. ACM, 17–28.

[64] X. Zhang and E. W. Knightly. 2016. WATCH: WiFi in active TV channels. *IEEE Trans. Cogn. Commun. Netw.* 2, 4 (2016), 330–342.

Received August 2020; revised January 2021; accepted January 2021