# Implementation of LPWAN over White Spaces for Practical Deployment

Mahbubur Rahman
Wayne State University, Detroit, MI, USA
r.mahbub@wayne.edu

Dali Ismail
Wayne State University, Detroit, MI, USA
dali.ismail@wayne.edu

Venkata P. Modekurthy
Wayne State University, Detroit, MI, USA
modekurthy@wayne.edu

Abusayeed Saifullah
Wayne State University, Detroit, MI, USA
saifullah@wayne.edu

## ABSTRACT

Low-Power Wide-Area Network (LPWAN) is an enabling Internet-of-Things (IoT) 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 the cost of licensed band, the recently proposed SNOW (Sensor Network over White Spaces) is a promising LPWAN platform that operates over the TV white spaces. Nevertheless, the current SNOW implementation uses USRP devices as LPWAN nodes which have high cost (≈ $750 USD per device) and large form-factor, hindering the applicability of this technology in practical deployment. In this paper, 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 CC1310) consequently brings down the cost and the form-factor of a SNOW node by 25x and 10x, respectively. Such implementation of SNOW on CC1310 devices 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 estimation, carrier frequency offset, and near-far power problem. Our deployment in the city of Detroit, Michigan demonstrates that CC1310-based SNOW can achieve uplink and downlink throughputs of 11.2kbps and 4.8kbps per node, respectively, over a distance of 1km. 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.*

## KEYWORDS

LPWAN, SNOW, White spaces, OFDM

**ACM Reference Format:**
Mahbubur Rahman, Dali Ismail, Venkata P. Modekurthy, and Abusayeed Saifullah. 2019. Implementation of LPWAN over White Spaces for Practical Deployment. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '19), April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3302505.3310080

## 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 [2], LTE-M [41], 5G [24]) or unlicensed/non-cellular (SNOW [34], LoRa [4], SigFox [35], etc.) bands. Most of the non-cellular technologies operate in the sub-1GHz ISM band except SNOW (Sensor Network over White Spaces) and WEIGHTLESS-W that operate in the TV white spaces [15].

*White spaces* are the allocated but locally unused TV spectrum (54-698MHz 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 [33]. 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 [15]) and SNOW [34] are the only two efforts to exploit the TV white spaces for LPWAN. Initially introduced in [32], SNOW is a highly scalable LPWAN technology offering reliable, bi-directional, concurrent, and asynchronous communication between a base station (BS) and numerous nodes [33, 34].

Despite its promise as a great LPWAN technology, SNOW has not yet received sufficient attention from the research community due to its limited availability for practical deployment. The current SNOW implementation, which is also available as open-source [37], 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 [30]. Using the USRP platform as the SNOW node limits the practical deployment of SNOW in real-world applications due to several factors including

*IoTDI '19, April 15–18, 2019, Montreal, QC, Canada*

ACM ISBN 978-1-4503-6283-2/19/04…$15.00
https://doi.org/10.1145/3302505.3310080

its high cost and large form-factor. As of today, 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, smart lighting, 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 [15].

In this paper, 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 that are deployable as SNOW nodes. Through this implementation, we empirically show that any COTS device with a programmable physical layer (PHY) that operates in the white spaces and supports amplitude-shift-keying (ASK) or binary phase-shift-keying (BPSK), can be practically deployed as SNOW nodes. Specifically, thanks to its programmable PHY, we use the widely available and low-power TI CC1310 [39] IoT device which costs approximately $30 USD (retail price) and is 10x smaller than a USRP B200 device (with antenna), thereby making SNOW adoptable for practical IoT applications.

The SNOW technology has never been implemented on IoT devices before. 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 [32–34]), which the implementation on CC1310 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 CC1310 device during downlink communication may be degraded 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 CC1310 device plays a vital role in both uplink and downlink communications. Without CSI estimation, the overall reliability and the communication range may be decreased. **Third**, Carrier frequency offset (CFO) needs to be handled robustly as the effects of CFO are much more pronounced in low-cost CC1310 devices, leading to severe inter-carrier-interference (ICI). ICI decreases the overall bitrate in both uplink and downlink communications of 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. Thus, we address the above challenge as well. Specifically, we make the following key technical contributions.

- We implement SNOW for practical deployment by programming the CC1310's PHY to work as SNOW nodes. Compared to the current USRP-based SNOW implementation, the cost and the form-factor of a single SNOW node are decreased approximately 25x and 10x, 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 CC1310 device to communicate directly with the SNOW BS from a distance of 1km. Additionally, we propose a pilot-based CFO estimation technique that takes into account the device mobility and increases reliability in both uplink and downlink communication. Finally, we address the near-far power problem in SNOW through an adaptive transmission power control (ATPC) protocol that improves the reliability in uplink communication.
- We experiment with the CC1310-based SNOW implementation through deployment in the city of Detroit, Michigan. Our results demonstrate that we achieve an uplink throughput of 11.2kbps per SNOW node. Additionally, our overall uplink throughput increases *linearly* with the increase in the number of SNOW nodes. In downlink, we achieve a throughput of 4.8kbps per SNOW node.

In the rest of the paper, Section 2 gives the system overview. Section 3 presents our SNOW implementation and how we address the associated practical challenges. Section 4 describes the near-far power problem and our ATPC mechanism. Sections 5 and 6 analyze the deployment cost and performance, respectively. Section 7 overviews related work. Section 8 concludes our paper.

## 2 BACKGROUND AND SYSTEM MODEL

### 2.1 An Overview of SNOW

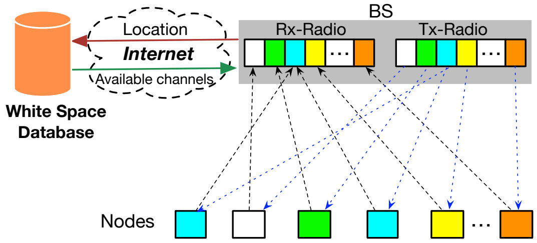


**Figure 1: Dual-radio BS and subcarriers [34]**

In this section, we provide a brief overview of SNOW. Its complete design and description is available in [34]. 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. The BS is powerful, Internet-connected, and line-powered while the nodes are power-constrained and do not have access to 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.

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 [14, 26, 28, 32–34]. Note that the SNOW PHY is different from MIMO radio design adopted in other wireless domains such as LTE, WiMAX, and 802.11n [33] 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). 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 MAC protocol similar to TinyOS [36]. Additionally, the nodes can autonomously transmit, remain in receive mode, or sleep. A node runs clear channel assessment (CCA) before transmitting. If its subcarrier is occupied, the node makes a random back-off in a fixed congestion back-off window. After this back-off expires, the node transmits immediately if its subcarrier is free. Then node repeats this operation until it sends the packet and gets the acknowledgment (ACK).

### 2.2 An Overview of TI CC1310

Texas Instruments introduced the TI CC1310 device as a part of the SimpleLink microcontroller (MCU) platform to support ultra-low-power and long-range communication [39]. With a small form-factor (length: 8cm, width: 4cm), CC1310 is designed to operate in the lower frequency bands (287–351MHz, 359–527MHz, and 718–1054MHz) including the TV band. The packet structure of the CC1310 device includes a `preamble`, followed by `sync word`, `length`, `payload`, and `CRC`, chronologically. It supports different data modulation techniques including Frequency Shift Keying (FSK), Gaussian FSK (GFSK), On-Off Keying (OOK), and a proprietary long-range modulation. It is capable of using a Tx/Rx bandwidth that ranges between 39 and 3767kHz. Additionally, with a supply voltage in the range of 1.8 to 3.8 volts, its Rx and Tx current consumption is 5.4mA and 13.4mA at +10dBm, respectively, offering ultra-low-power communication. Overall, it offers wide availability at low-cost, long-range communication support, a variety of data modulations, and the ability to program and reconfigure itself.

## 3 SNOW IMPLEMENTATION ON TI CC1310

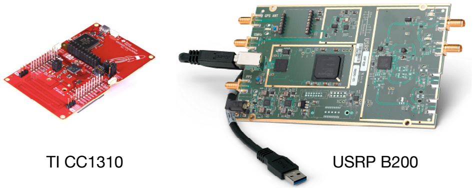


**Figure 2: Devices used in our SNOW implementation. A node is a CC1310 device. The BS has two USRP B200 devices, each having its own antenna. Antenna is not shown in this figure, which is approximately 2x bigger than the USRP device.**

The original SNOW implementation in [34] uses the USRP hardware platform for both the BS and the nodes. In our implementation, we use the CC1310 device as SNOW nodes and USRP in the BS (Figure 2). For BS implementation, we adopt the open-source code provided in [37]. 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 platform that gives a high magnitude of freedom to perform baseband signal processing [7]. In the following, we explore a number of implementation considerations and feasibility for a CC1310 device as a SNOW node for practical deployments. First, we show how to configure a CC1310 device to make it work as a SNOW node. We then address the practical challenges (e.g., PAPR problem, CSI and CFO estimation) associated with our CC1310-based SNOW implementation.

### 3.1 Configuring TI CC1310

To configure the subcarrier center frequency, bandwidth, modulation, and the Tx power we set the appropriate values to the CC1310 command inputs `centerFreq, rxBw, modulation`, and `txPower`, respectively, using *Code Composer Studio* (CCS) provided by Texas Instruments [39]. A graphical user interface alternative to CCS is *SmartRF Studio*, which is also provided by Texas Instruments. The MAC protocol of SNOW in CC1310 is implemented on top of the example CSMA/CA project that comes with CCS.

### 3.2 Peak-to-Average Power Ratio Observation

By transmitting on a large number of subcarriers simultaneously (in downlink), the BS suffers from a traditional OFDM problem called *peak-to-average power ratio (PAPR)*. PAPR of an OFDM signal is defined as the ratio between the maximum instantaneous power and its average power. In SNOW downlink communication (i.e., BS to nodes), after the IFFT is performed by the BS, the composite signal can be represented as follows.

$$x(t) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k \, e^{j2\pi f_k t}, \; 0 \le t \le NT$$

Here, $X_k$ is the modulated data symbol for node $k = \{0, 1, \cdots, N-1\}$ on subcarrier center frequency $f_k = k\Delta f$, where $\Delta f = \frac{1}{NT}$ and $T$ is the symbol period. Therefore, the PAPR can be calculated as [18]

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

Here, the average power $P_{\text{avg}} = E[|x(t)|^2]$. 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 (DAC) 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, the out-of-band power will exceed the specified limit and introduce severe ICI [18]. Moreover, the in-band distortion (constellation tilting and scattering) due to high PAPR may cause severe performance degradation [19]. It has been shown that the PAPR reduction results in significant power saving at the transmitters [5].

As shown in Figure 3, the PAPR in SNOW downlink communication (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

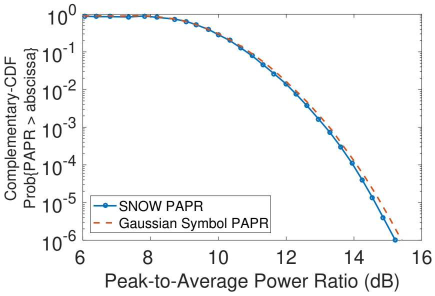


**Figure 3: PAPR distribution of D-OFDM signal in Tx-Radio.**

back-off (IBO) [5] equivalent to the PAPR at a probability of $10^{-4}$. Here, PAPR ≈ 14dB or 25.12. Thus, the efficiency ($\eta = 0.5/\text{PAPR}$) of the HPA [18] 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 [18]). 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 [34], or any other node's data. Thus, none of the existing PAPR reduction techniques will work in our implementation.

To this extent, we address the PAPR problem in SNOW by allocating a special subcarrier called *downlink subcarrier* for downlink 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. If the BS requires (downlink subcarrier is being interfered by an external source), it may allocate several redundant downlink subcarriers. Note that the dual-radio architecture in SNOW 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 $i$, it will set the $i$-th bit in the bit-vector. Upon receiving the bit-vector, that node may get the acknowledgment by looking at the $i$-th bit of the vector. A node retransmits the packet if that packet is not acknowledged in the first valid ACK received by that node. In the following, we describe our technique to handle a **rare** case in practical SNOW deployments, and hence may be kept optional in implementation.

When a subcarrier (say, $i$) is shared by multiple nodes, the BS may receive a valid second packet (say, from node A) before transmitting the ACK for the valid first packet (say, from node B). In this case, both nodes A and B may be acknowledged by setting the $i$-th bit of the vector. However, if the packet from node A (or, B) is valid and the packet from node B (or, A) is invalid, the BS will reset the $i$-th bit of the vector and transmit the ACK. Thus, none of the packets are acknowledged even if one of them is valid. To compensate for that, the BS (Tx-Radio) will switch to node A's (or, B's) subcarrier and transmit an ACK packet. Thus, in our implementation, if a node finds that its packet is not acknowledged in the first valid ACK it received, before retransmission it listens to its subcarrier for a fixed amount of time. Each node may know this fixed time when it joins the network. Typically, if a subcarrier is shared by $G$ nodes, the fixed amount of time (worst case) may be set to $GD_p$ (ignoring the frequency switching time in the Tx-Radio), where $D_p$ is the time to transmit one packet. Other ways of addressing such issue may include the use of *hash functions*. However, we do not explore that in our implementation for scalability issue due to hash collision.

## 3.3 Does Channel State Information Estimation Make It More Resilient?

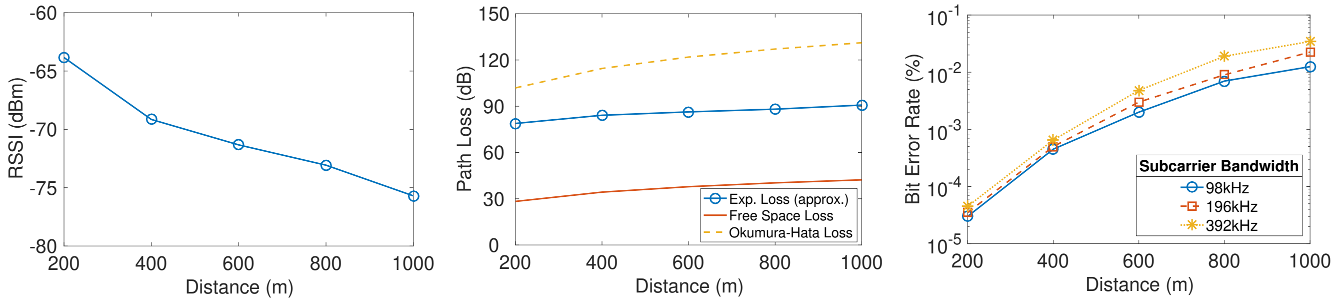
Multi-user OFDM communication requires channel estimation and tracking for ensuring high data rate at the BS. One way of avoiding channel estimation is to use the *differential phase-shift keying (DPSK)* modulation technique. However, the use of DPSK results in a lower bitrate at the BS due to a 3dB loss in the singal-to-noise ratio (SNR) [43]. 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 uplink. Thus, it requires handling of the channel estimation in SNOW.

Figure 4 shows the received signal strength indicator (RSSI), path loss, and BER at the SNOW BS for a CC1310 device that transmits from 200 to 1000m distances with a Tx power of 15dBm, subcarrier center frequency at 500MHz, and a bandwidth of 98kHz. 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 [29]. We also compare with the Okumura-Hata [29] 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 [16]) beyond 400m 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 [40], where the channel conditions vary slowly with respect to the duration of a single node–BS packet duration. Note that joint-CSI estimation [17, 21, 31] in SNOW is not our design goal since it would require SNOW nodes to be strongly time-synchronized. Similar to IEEE 802.16e [11], 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 other nodes as well. 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 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



(a) RSSI vs distance

(b) Path Loss vs distance

(c) BER vs distance

**Figure 4: RSSI, path loss, and BER at the SNOW BS for a TI CC1310 node from different distances.**

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). In our case, n = 4 which yields similar complexity for CSI estimation in IEEE 802.11 [9].

Let, the preamble sequence be $(p_1, p_2, \cdots, p_n)$, where vector $p_i$, for $i = \{1, \cdots, n\}$, is transmitted as

$$y_i = Hp_i + w_i.$$

Combining the received preamble sequences, we get

$$Y = [y_1, \cdots, y_n] = HP + W.$$

Here, $P = [p_1, \cdots, p_n]$ and $W = [w_1, \cdots, w_n]$. With combined knowledge of $Y$ and $P$, channel matrix $H$ is estimated. Similar to the CSI estimation in the uplink communication by the BS, each node also calculates the CSI estimation in downlink communication.

## 3.4 Does Carrier Frequency Offset Estimation Make It More Robust?

Multi-user OFDM systems are very much 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; (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.

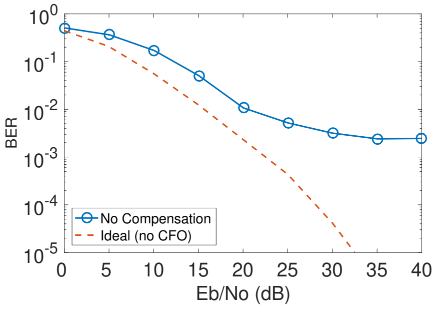


**Figure 5: BER at different $E_b/N_0$.**

The loss in SNR due to the CFO between the SNOW BS and a node can be estimated as: $SNR_{loss} = 1 + \frac{1}{3}(\pi\delta fT)^2\frac{E_s}{N_0}$, where $\delta f$ 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. To show the CFO effects, we choose two neighboring orthogonal subcarriers in the BS and send packets from two nodes. Figure 5 shows the BER at the BS from those two nodes at different $E_b/N_0$, where $E_b$ is the average energy per bit in the received signals. This figure shows that BER is nearly $10^{-3}$ even for very high $E_b/N_0$ ($\approx$ 40dB), which is also very high compared to the theoretical BER [6]. 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 CFO estimation, it is not robust and does not account for mobility of the nodes [34]. We propose a pilot-based CFO estimation technique that is robust and accounts for the node's mobility. 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 $f_{\text{node}}$ and $f_{\text{BS}}$ 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 $\delta f$ is given by $y(t) = x(t)e^{j2\pi\delta ft}$. Similar to IEEE 802.11a [9], we estimate $\delta f$ 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 $\delta t_s$ as the short preamble duration and $\delta f_s$ 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

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

Taking angle of both sides gives us as follows.

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

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 $\delta f_s$, we correct each time domain sample (say, $P$) received in the long preamble as $P_a = P_a e^{-ja\delta f_s}$, where $a = \{1, 2, \cdots, A\}$ 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 $\delta t_l$ 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 *ppm (parts per million)* on the BS's crystal is given by $\text{ppm}_{\text{BS}} = 10^6\left(\frac{\delta f}{f}\right)$. Thus, the BS calculates $\delta f_i$ 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 [45]. Thus, BS also knows the CFO for downlink communication.

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 [38]. 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 $\delta f_d$ as the CFO due to the Doppler shift, $v$ as the velocity of the node, and $\theta$ as the angle of the arrived signal at the BS from the node, we have [38]

$$\delta f_d = f_i\left(\frac{v}{c}\right)\cos(\theta).$$

Here, $f_i$ 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 $\delta s$ is the amount of anticipated change in nodes position during a packet transmission and $r$ is the *line-of-sight* distance of the node from the BS. Thus, CFO compensation due to the Doppler shift is done at the nodes during uplink communication. In downlink communication, the Tx-Radio of the BS can also compensate for the node's mobility as the node can report its Doppler shift to the BS during uplink communication.

In summary, as the nodes asynchronously transmit to the BS, doing the joint-CFO estimation for each subcarrier at the BS is quite difficult. Thus, we use a simple feedback approach for proactive CFO correction in uplink communication. $\delta f_i$ estimated at the BS for subcarrier $f_i$ is given to the node (during its joining process) that is assigned subcarrier $f_i$. The node may then adjust its transmitted signal based on $\delta f_i$ and $\delta f_d$, 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 communication. Such feedback-based proactive compensation scheme was studied before for multi-user OFDM [42] and is also used in global system for mobile communication (GSM).

# 4 NEAR-FAR POWER PROBLEM

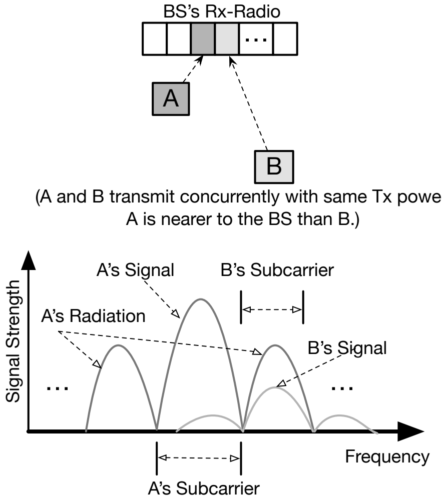
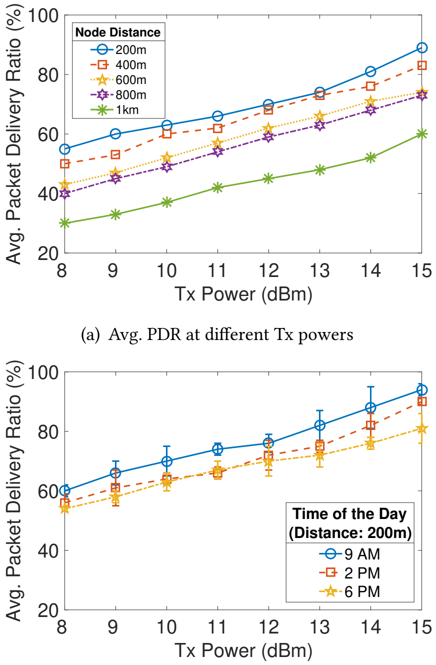


**Figure 6: Illustration of near-far power problem.**

Wireless communication is susceptible to the near-far power problem, especially in CDMA [23]. Multi-user D-OFDM system in SNOW may also suffer from this problem. We give an illustration of the near-far power problem in SNOW in Figure 6. 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 in the bottom 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.

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 nodes within 20m of the BS that use the left and the right subcarrier, respectively. We use another node that uses the middle subcarrier and is placed at different distances between 200 and 1000m from the BS. Nodes that are within 20m of the BS transmit packets continuously with a transmission power of 0dBm. At each distance, for each transmission power between 8 and 15dBm, the node that uses the middle subcarrier sends 100 rounds of 1000 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 0-500ms. For each transmission power level, at each distance, that node calculates its average *packet delivery ratio (PDR)*. We repeat the same set of 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 200m) and shows that the average PDR changes at different time of the day. Overall, Figure 7(a) and 7(b) confirms that the average PDR increases with the increase in the transmission power. Thus, the near-far power problem needs



(a) Avg. PDR at different Tx powers

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

**Figure 7: Packet deliery ratio at different Tx powers**

to be addressed in SNOW. To this extent, we propose an adaptive transmission power control for SNOW design.

## 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 [20] in fundamental concepts of the network design and architecture. In [20], the authors have considered a multi-hop wireless sensor network based on IEEE 802.15.4 [10] 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 [20]. In the following, we describe our model.

Whenever a node is assigned a new subcarrier, changes location (inside the SNOW network), or observes a lower PDR, e.g., PDR below quality of service (QoS) requirements, 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. Thus, our predictive model uses two vectors: $TP$ and $L$, where $TP = \{tp_1, tp_2, \cdots, tp_m\}$ contains $m$ different Tx power levels that the node uses to send $m$ groups of packets to the BS and $L = \{l_1, l_2, \cdots, l_m\}$ 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 as follows.

$$a \sum (tp_i)^2 + b \sum tp_i = \sum l_i . tp_i$$
$$a \sum tp_i + b \; m = \sum l_i$$

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 $a$ is assumed time-invariant in the predictive model. On the other hand, 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 [20].

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

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

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

Here, $l_k(t-1)$, for $k = \{1, 2, \cdots, K\}$, 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:

$$tp(t) = \Big[\frac{PDR_{\text{threshold}} - \hat{b}(t)}{\hat{a}}\Big].$$

## 5 PRACTICAL DEPLOYMENT ARCHITECTURE AND COST OF SNOW

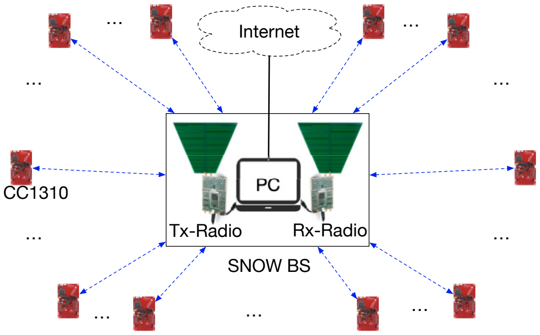


**Figure 8: SNOW Network. PC may be replaced by a Raspberry Pi device. Two USRP B200 devices may be replaced by a USRP B2100 device that has two half-duplex radios.**

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 which has two half-duplex radios. Also, a Raspberry Pi [25] device may be used instead of the PC. All the CC1310 nodes are battery-powered and directly connected to the BS.

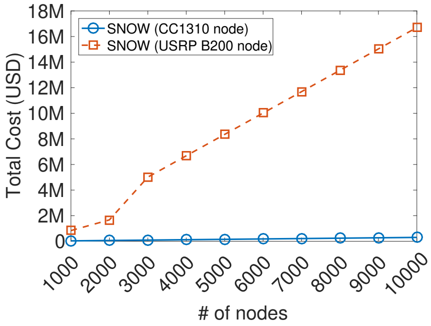


**Figure 9: Practical deployment cost with numerous nodes.**

We now analyze the deployment cost of our CC1310-based SNOW implementation and compare with the original USRP-based SNOW implementation in [34]. Figure 9 shows the total deployment cost of our CC1310-based SNOW implementation for different numbers of nodes between 1000 and 10,000. A CC1310 device costs approximately $30 USD (retail price). 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 [34], 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. Figure 9 shows that to deploy a SNOW network with 1000 nodes, the CC1310-based SNOW implementation may cost approximately $31.6K USD, compared to $836.6K USD for the USRP-based implementation in [34]. For a SNOW deployment of 5000 nodes, the costs are $151.6K and $8.4M for CC1310-based implementation and USRP-based implementation, respectively; for a deployment of 10,000 nodes, the costs are $301.6K and $16.7M, respectively. Our new implementation of SNOW on the CC1310 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 of our CC1310-based SNOW implementation. We evaluate both uplink and downlink performances with both stationary and mobile CC1310 nodes.

### 6.1 Setup

Figure 10 shows our deployment in the city of Detroit, Michigan. We deploy 25 CC1310 devices at different distances between 200 and 1000m. We use the TV white space channel with frequency band 500–506MHz and split into 29 (numbered 1 to 29) overlapping (50%) orthogonal subcarriers, each 400kHz wide. Note that the USRP-based SNOW also uses a similar subcarrier bandwidth [34]. We use the 28th subcarrier as join subcarrier and the 26th subcarrier as downlink subcarrier. We do not use the 29th and the 27th subcarriers such that the join subcarrier may remain ICI-free (Section 3.4). The remaining 25 subcarriers are assigned to different nodes. We use the packet structure of CC1310 (`preamble` (32 bits), `sync word` (32 bits), `length`, `payload`, and `CRC` (16 bits)). Our default payload length is 30 bytes and contains random data. Our default bandwidth at the CC1310 nodes is 39kHz. We use OOK modulation supported by the CC1310 device. Unlike the USRP-based SNOW, we do not use any spreading factor. Since the subcarrier bandwidths at the BS and the CC1310 nodes are 400kHz and 39kHz, 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 15dBm. However, a CC1310 device may choose to operate with any transmission power between 0 and 15dBm, as needed by our ATPC model (Section 4.1). The receive sensitivity at the BS is set to -114dBm, as per the white space regulations [27]. Unless stated otherwise, these are the default parameter settings.

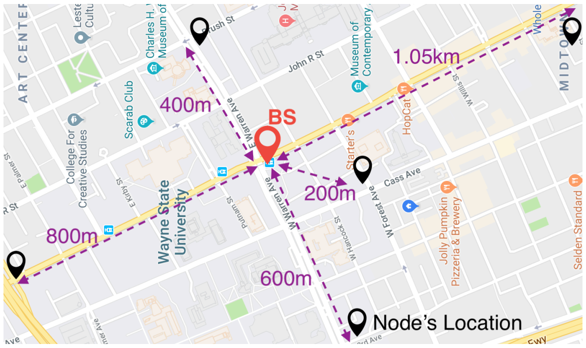
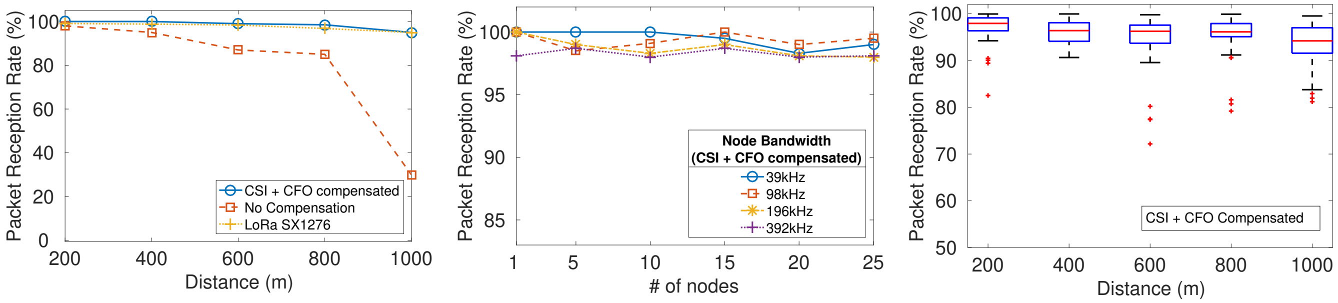


**Figure 10: SNOW deployment in Detroit, Michigan.**



(a) Packet reception rate at a node at different distances (b) Packet reception rate in uplink communication (c) Packet reception rate in downlink communication

**Figure 11: Reliability in long distance communication.**

## 6.2 Reliability over Long Distance

*6.2.1 Achievable Distance.* We first test the achievable communication range of our CC1310-based SNOW implementation. We take one CC1310 device and transmit to the BS from different distances between 200 and 1000m. We keep our antenna height at 3 meters above the ground for both the BS and the node. At each distance, the CC1310 transmits 1000 packets with a random interval between 0 and 500ms. The transmission power is set to 15dBm. 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: 125kHz, spreading factor: 7, coding rate: 4/5) with the above settings. 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. As shown in this figure, with CSI and CFO compensation, the BS achieves 95% of PRR from a distance of 1km . Without CSI and CFO compensation, the PRR at the BS is as low as 30% from 1km distance. This figure also shows that a LoRa SX1276 device can deliver packets to another over 1km with a PRR of 95%, which is similar to the CC1310-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 1km, PRR stars decreasing in our implementation. Our best guess is that if we can place the BS or the node at a higher altitude (FCC allows up to 30 meters), we may achieve high reliability over much longer communication range.

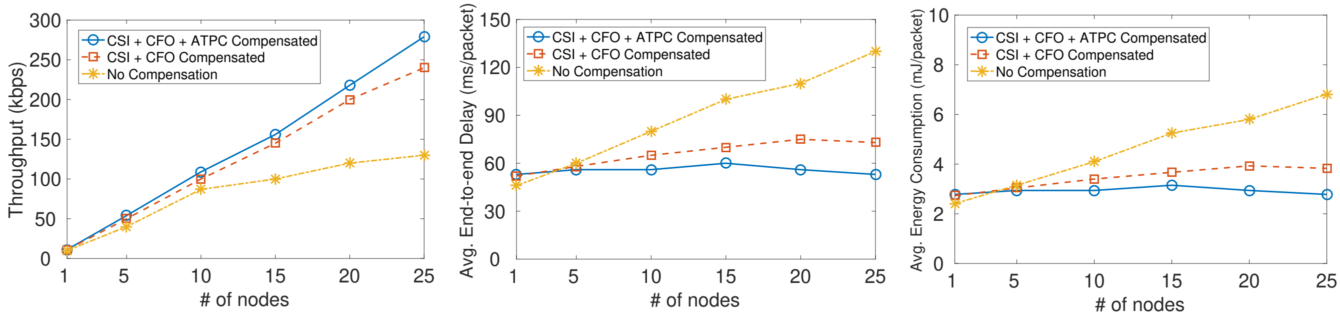
*6.2.2 Uplink Reliability.* To show the uplink reliability under concurrent transmissions from different nodes (CFO and CSI compensated), we transmit from 1 to 25 nodes (using their assigned subcarriers) to the BS. In this experiment, all the nodes are distributed within 200 and 1000m of the BS. Each node uses different subcarrier bandwidths between 39 and 392kHz. For each bandwidth starting from 39kHz, a node sends consecutive 1000 packets. Between each bandwidth, a node sleeps for 500ms. 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% reliability when 25 nodes transmit concurrently using 39kHz, and up to 98% using 392kHz. Thus, ensuring high uplink reliability of our CC1310-based implementation over long distances.

*6.2.3 Downlink Reliability.* In 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 1000m of the BS. We repeat the same experiment 50 times with an interval between 0 and 500ms. 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 downlink is as high as 99% for 75% of the nodes that are approximately 200m away from the BS. Also, 75% of the nodes that are approximately 1km away from the BS achieve a PRR of 95%. Thus, this experiment confirms high downlink reliability of our CC1310-based implementation over long distances.

## 6.3 Performance in Uplink Communication

In this section, we evaluate the uplink network performance in terms of throughput, end-to-end (E2E) delay, and energy consumption. We calculate the throughput at the BS as the total achieved bitrate (kbps). We estimate the E2E delay at the nodes as the time (ms) elapsed between a packet transmit and its ACK receive. We also measure the energy consumption (mJoule) at the nodes. We allow from 1 to 25 nodes to transmit concurrently to the BS. We distribute the nodes between 200 and 1000m in our testbed. Each node transmits 1000 30-byte (payload length) packets with a random packet interval between 0 and 100ms. Such packet interval confirms that the node's transmissions are indeed asynchronous to the BS. Each node uses a bandwidth of 39kHz. 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. Thus, a node may try several times to deliver a packet to the BS.

*6.3.1 Throughput.* Figure 12(a) shows that the BS achieves up to 279kbps of throughput when 25 nodes transmit concurrently (case 1), yielding 11.16kbps per node. Additionally, the overall throughput at the BS increases linearly with an increase in the number of nodes. When only CSI and CFO are compensated for, the overall throughput at the BS also increases with an increase in the number



(a) Throughput vs # of nodes

(b) End-to-end delay vs # of nodes

(c) Energy consumption vs # of nodes

**Figure 12: Network performance in uplink communication.**

of concurrent transmissions, however, it depends on the nodes' distribution (physical) across the network. If there is no near-far power problem, the overall throughput may be the same as observed in case 1. With no compensation, the achieved throughput per node is approximately 5kbps, thus 2x lesser than case 1. Note that a CC1310 device can generate a baseband signal with a symbol rate of 11.2kbaud (OOK modulated). Thus, using a node bandwidth of 39kHz or 392kHz 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. Note that if we use any other COTS device that can generate a higher symbol rate for OOK at higher node bandwidth, the per node throughput may also increase with an increase in the node bandwidth. Overall, CC1310-based SNOW implementation shows high potential for practical deployments.

*6.3.2 End-to-end Delay.* Figure 12(b) shows the average E2E delay per packet at the nodes. When CSI, CFO, and ATPC are compensated for, the average E2E delay per packet in the network is 55ms with 25 concurrent transmissions. Also, for case 1, the average E2E 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 E2E delay per packet increases a little bit with an increase in the number of concurrent transmissions. With no compensation, the average E2E 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 CC1310-based SNOW implementation shows great promise for low-latency Industry 4.0 applications [22].

*6.3.3 Energy Consumption.* Figure 12(c) shows the average energy consumption per packet at the nodes. We use the CC1310 energy profile to calculate the energy consumption during Tx, Rx, and idle time [39]. For case 1, where the CSI, CFO, and ATPC are compensated for, the average energy consumption per packet in the network is approximately 2.78mJoule with 25 concurrent transmissions. Also, the average energy consumption per packet 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 packet increases to 3.83mJoule 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 the CC1310-based SNOW may host long-lasting IoT applications.

## 6.4 Performance in Downlink Communication

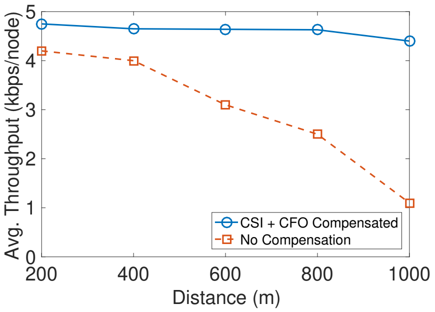
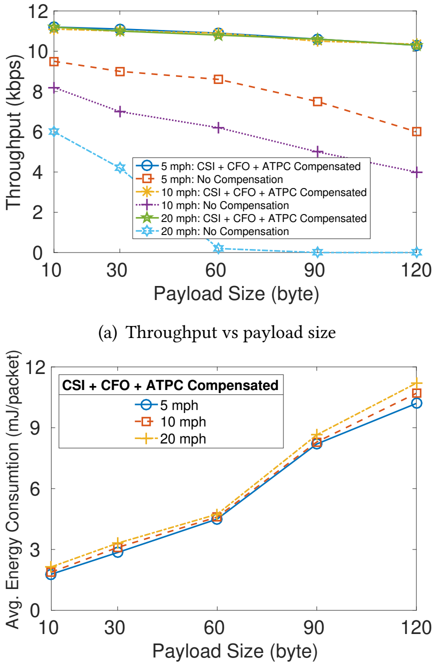


**Figure 13: Throughput in downlink communication.**

In this section, we evaluate the downlink network performance in terms of throughput. The BS sends 1000 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 downlink, the BS uses a bandwidth of 39kHz and sends packets to the nodes using the Tx-radio. 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 200m away from the BS can achieve an average downlink throughput of 4.8kpbs, 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 CC1310 device can successfully receive an OOK-modulated signal with 4.8kbaud symbol rate and 39kHz bandwidth [39]. Overall, our CC1310-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 CC1310 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 600m far from the BS and



(a) Throughput vs payload size

(b) Energy consumption vs payload size

**Figure 14: Throughput and energy consump. under mobility.**

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 50ms, using their assigned subcarriers, each 39kHz wide. We vary the speed of the mobile node approximately to 5mph, 10mph, and 20mph in any arbitrary direction within our network range. At each speed, we change the payload size of the mobile node between 10 and 120bytes. For each payload size, the mobile node transmits to the BS 1000 packets with a random interval between 0 and 50ms. 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.

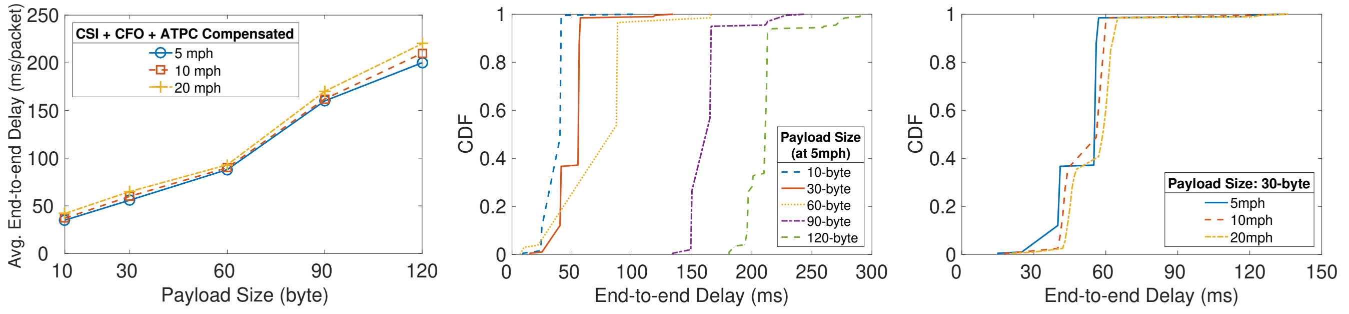
*6.5.1 Throughput.* Figure 14(a) shows the throughput at the BS (of the mobile node) for different speeds and payload sizes. As this figure suggests, the throughput decreases slightly from 11.18kbps to 10.3kbps at 5mph, 10.35kbps at 10mph, and 10.3kbps at 20mph for an increase in the payload size between 10 and 120bytes, 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 20mph, the throughput drops to approximately 0 for payload size of 60bytes. In general, the packet size is susceptible to node's mobility. In fact, if CSI and CFO are not compensated for, 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 node's mobility.

*6.5.2 Energy Consumption.* Figure 14(b) shows that the average energy consumption per packet increases slightly higher than linear with an increase in the payload size, when CSI, CFO, or/and ATPC are compensated for. For example, at 5mph, it takes on average 1.78mJoule, 2.85mJoule, 4.5mJoule, 8.2mJoule, and 10.2mJoule to transmit a payload of size 10, 30, 60, 90 and 120bytes, respectively. Also, the average energy consumption per packet increases with an increase in the speed. As shown in this figure, the average energy consumption per packet is approximately 1.78mJoule at 5mph and 2.14mJoule at 20mph, for a payload of size 10bytes. Our best guess is that at higher speeds the mobile node retransmits several packets multiple times due to ACK loss, high BER at BS, or/and ATPC. Overall, Figure 14(b) confirms that our CC1310-based SNOW implementation is energy efficient under node's mobility.

*6.5.3 End-to-end Delay.* Figure 15(a) shows that the average E2E delay per packet at the mobile node increases with an increase in speed and payload size. For example, at 5mph, the average E2E delay per packet with a payload of size 10, 30, 60, 90, and 120bytes are 35, 56, 88, 160, 200ms, respectively; at 10mph, the average E2E delays are 37, 60, 90, 162, 210ms, respectively; at 20mph, the average E2E delays are 42, 65, 93, 170, 220ms, respectively. Moreover, Figure 15(b) shows the cumulative distribution function (CDF) of the E2E delay at a constant speed of 5mph with varying payload sizes. This figure shows that 60% of the 10-byte (payload length) packets observe an E2E delay more than 35ms, 65% of the 30-byte (payload length) packets observe an E2E delay more than 55ms, 50% of the 60-byte (payload length) packets observe an E2E delay more than 90ms, 98% of the 90-byte (payload length) packets observe an E2E delay more than 150ms, and 95% of the 120-byte (payload length) packets observe an E2E delay more than 195ms. Furthermore, Figure 15(c) shows the CDF of E2E delays for a fixed payload length of 30bytes at varying speed. As this figure shows, 98% of the packets at 5mph observe an E2E delay up to 55ms, 99.99% of the packets at 10mph observe an E2E delay up to 60ms, and 98% of the packets at 20mph observe an E2E delay up to 65ms. Overall, Figure 15 confirms that our CC1310-based SNOW implementation may provide very low latency under node's mobility.

## 7 RELATED WORK

Recently, a number of LPWAN technologies have been developed that operate in the licensed (LTE-M [41], NB-IoT [2], EC-GSM-IoT [8], 5G [1]) or unlicensed (LoRa [4], SigFox [35], INGENU [12], IQRF [13], DASH7 [3], WEIGHTLESS-SIG [44]) spectrum. Operating in the licensed band is costly due to high service fee and costly infrastructure. On the contrary, most non-cellular LPWANs, except SNOW and WEIGHTLESS-W, operate in the ISM band. While the ISM band is unlicensed, it is getting 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 widely available, less crowded, and wide spectrum of the TV white spaces. In contrast to the numerous works that mostly focused on exploiting the white spaces for broadband access (see survey [27]), SNOW exploited white spaces for LPWAN. 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 paper focuses on implementing SNOW using the cheap and widely available COTS devices for practical and scalable deployment of SNOW.



(a) End-to-end delay vs payload size (b) CDF of end-to-end delay at different payload sizes (c) CDF of end-to-end delay at different speeds

**Figure 15: End-to-end delay under node's mobility.**

## 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 paper, we have implemented SNOW for practical deployments using the COTS device CC1310 as SNOW nodes. Our CC1310-based SNOW implementation decreases the cost and the form-factor of a single SNOW node by 25x and 10x, respectively. We have also addressed several practical deployment challenges that include PAPR, CSI and CFO estimation, and near-far power problem. We have deployed the CC1310-based SNOW in the city of Detroit, Michigan and achieved per node uplink and downlink throughputs of 11.2kbps and 4.8kbps, respectively, over a distance of 1km. Additionally, our overall uplink throughput at the BS have increased linearly with the increase in the number of nodes. Our extensive experiments have demonstrated the CC1310-based SNOW as a feasible LPWAN technology that can be deployed practically at low-cost and in large-scale for future IoT applications.

## ACKNOWLEDGMENTS

This work was supported by NSF through grant CNS-1742985.

## REFERENCES

[1] 3GPP. 2014. The LTE STANDARD. https://www.qualcomm.com/media/documents/files/the-lte-standard.pdf.
[2] 3GPP. 2018. NBIoT. 3gpp.org/news-events/3gpp-news/1785-nb_iot_complete.
[3] Dash Alliance. 2018. DASH7. http://www.dash7-alliance.org.
[4] LoRa Alliance. 2018. LoRaWAN. https://www.lora-alliance.org.
[5] R. J Baxley and G T. Zhou. 2004. Power savings analysis of peak-to-average power ratio in OFDM. *IEEE Trans. on Consumer Electronics* 50, 3 (2004), 792–798.
[6] J. Choi, Y-H Lee, C. Lee, and H W Jung. 2000. Carrier frequency offset compensation for uplink of OFDM-FDMA systems. In *ICC '00*. IEEE, USA, 425–429.
[7] GNURadio. 2018. GNURadio. http://gnuradio.org.
[8] GSMA. 2018. GSMA IoT. https://www.gsma.com/iot/.
[9] IEEE. 2018. IEEE 802.11. http://www.ieee802.org/11.
[10] IEEE. 2018. IEEE 802.15.4. http://standards.ieee.org/about/get/802/802.15.html.
[11] IEEE. 2018. IEEE 802.16e. http://www.ieee802.org/16/tge/.
[12] Ingenu. 2017. RPMA. https://www.ingenu.com/technology/rpma.
[13] IQRF. 2018. IQRF. http://www.iqrf.org/technology.
[14] D. Ismail, M. Rahman, and A. Saifullah. 2018. Demo Abstract: Implementing SNOW on Commercial Off-The-Shelf Devices. In *IoTDI '19*. IEEE, USA, 2.
[15] D. Ismail, M. Rahman, and A. Saifullah. 2018. Low-power wide-area networks: opportunities, challenges, and directions. In *ICDCN '18*. ACM, NY, USA, 8.
[16] D. Ismail, M. Rahman, A. Saifullah, and S. Madria. 2017. RnR: Reverse & Replace Decoding for Collision Recovery in Wireless Sensor Networks. In *SECON '17*. IEEE, USA, 9.
[17] M. Jiang, J. Akhtman, and L. Hanzo. 2007. Iterative joint channel estimation and multi-user detection for multiple-antenna aided OFDM systems. *IEEE Trans. on Wireless Comm.* 6, 8 (2007), 1–11.
[18] T. Jiang and Y. Wu. 2008. An overview: Peak-to-average power ratio reduction techniques for OFDM signals. *IEEE Trans. on broadcasting* 54, 2 (2008), 257–268.
[19] B. Kamali, R. A. Bennett, and D. C. Cox. 2012. Understanding WiMAX: an IEEE-802.16 standard-based wireless technology. *IEEE Potentials* 31, 5 (2012), 23–27.
[20] 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. on Sensor Networks* 12, 1 (2016), 6.
[21] L Maniatis, T Weber, A Sklavos, and Y Liu. 2002. Pilots for joint channel estimation in multi-user OFDM mobile radio systems. In *ISSSTA '02*. IEEE, USA, 5.
[22] V. P. Modekurthy, D. Ismail, M. Rahman, and A. Saifullah. 2018. A Utilization-Based Approach for Schedulability Analysis in Wireless Control Systems. In *ICII '18*. IEEE, USA, 10.
[23] A. Muqattash and M. Krunz. 2003. CDMA-based MAC protocol for wireless ad hoc networks. In *MobiHoc '03*. ACM, NY, USA, 12.
[24] ngmm. 2018. ngmn. http://www.ngmn.org.
[25] Raspberry pi. 2018. Raspberry pi. https://www.raspberrypi.org/.
[26] M. Rahman, D. Ismail, and A. Saifullah. 2018. Demo Abstract: Enabling Inter-SNOW Concurrent P2P Communications. In *IoTDI '19*. IEEE, USA, 2.
[27] M. Rahman and A. Saifullah. 2018. A Comprehensive Survey on Networking over TV White Spaces. arXiv:cs.NI/1810.07120
[28] M. Rahman and A. Saifullah. 2018. Integrating Low-Power Wide-Area Networks in White Spaces. In *IoTDI '18*. IEEE, USA, 6.
[29] T. S Rappaport et al. 1996. *Wireless communications: principles and practice*. Vol. 2. Prentice Hall PTR, New Jersey, USA.
[30] Ettus Research. 2018. USRP B2100. https://www.ettus.com/product/.
[31] C. Ribeiro, M J F-G Garcia, V P G Jiménez, A Gameiro, and A Armada. 2008. Uplink channel estimation for multi-user OFDM-based systems. *Wireless Personal Communications* 47, 1 (2008), 125–136.
[32] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu. 2016. SNOW: Sensor Network over White Spaces. In *SenSys '16*. ACM, NY, USA, 14.
[33] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra. 2017. Enabling Reliable, Asynchronous, and Bidirectional Communication in Sensor Networks over White Spaces. In *SenSys '17*. ACM, NY, USA, 14.
[34] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra. 2018. Low-Power Wide-Area Network Over White Spaces. *ACM/IEEE Transactions on Networking* 26, 4 (2018), 1893–1906.
[35] SigFox. 2018. SigFox. http://sigfox.com.
[36] SING. 2018. TinyOS. http://www.tinyos.net.
[37] SNOW. 2018. Base Station. https://github.com/snowlab12/gr-snow.
[38] S. L Talbot and B. Farhang-Boroujeny. 2007. Mobility and carrier offset modeling in OFDM. In *GLOBECOM'07*. IEEE, USA, 5.
[39] TI. 2018. CC1310 Chip. http://www.ti.com/tool/launchxl-cc1310.
[40] D. Tse and P. Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press, Cambridge, UK.
[41] UBlox. 2018. LTE-M. https://www.u-blox.com/en/lte-cat-m1.
[42] J-J Van de Beek, P. O. Borjesson, M-L Boucheret, D. Landstrom, J. M. Arenas, P. Odling, Christer Ostberg, Mattias W., and S. K. Wilson. 1999. A time and frequency synchronization scheme for multiuser OFDM. *IEEE Journal on Selected Areas in Comm.* 17, 11 (1999), 1900–1914.
[43] J-J Van De Beek, O. Edfors, M. Sandell, S K Wilson, and P O Borjesson. 1995. On channel estimation in OFDM systems. In *VTC '95*. IEEE, USA, 815–819.
[44] Weightless. 2018. Weightless. http://www.weightless.org.
[45] Y. Yao and G. B Giannakis. 2005. Blind carrier frequency offset estimation in SISO, MIMO, and multiuser OFDM systems. *IEEE Trans. on Comm.* 53, 1 (2005), 173–183.