

Enabling Massive Scalability in Low-Power Wide-Area Networks

Mahbubur Rahman

Computer Science, Graduate Center and Queens College
City University of New York, USA

Pushpen Bikash Goala

Computer Science, Graduate Center
City University of New York, USA

Abstract—Low-power wide-area networks (LPWANs) have become ubiquitous in the Internet of Things (IoT) applications due to their ability to connect sensors across large geographic areas over a single hop. It is, however, very challenging to achieve massive scalability in LPWANs where numerous sensors can transmit data energy efficiently and with low latency, which may be required by the emerging IoT applications. In this paper, we address the above challenges by significantly advancing an LPWAN technology called SNOW (sensor network over white spaces). SNOW exploits D-OFDM (distributed orthogonal frequency division multiplexing) to enable parallel reception of data to a base station (BS) from multiple asynchronous sensors, each sensor using a different D-OFDM subcarrier. In this paper, we achieve massive scalability in SNOW by enabling the BS to decode parallel data from multiple asynchronous sensors on the same subcarrier while concurrently decoding from other subcarriers as well. To do this, we develop a set of pseudorandom noise (PN) sequences that are mutually non-interfering within and across the D-OFDM subcarriers. Each sensor uses its own PN sequence (from the set) for modulating data on its subcarriers, enabling massive concurrency in SNOW. Our simulation results show that we can achieve approximately 9x more scalability in SNOW while being both energy efficient at the sensors and timely in data collection at the BS, thereby enabling emerging IoT applications that require longer sensor battery life as well as make time-critical, data-driven decisions.

Index Terms—LPWAN, SNOW, OFDM, spread spectrum.

I. INTRODUCTION

The number of Internet of Things (IoT) connections is expected to reach 83 billion by 2024, with an industry value of over a trillion dollars. The emerging IoT and CPS (cyber-physical systems) applications, including sensing and monitoring, smart farming, and oil field management aim to utilize IoT devices for enhancing sustainability, quality of life, health, safety, and economic prosperity of communities in both urban and rural areas. IoT devices (i.e., sensors or simply nodes) are usually battery-powered, scattered in large numbers (e.g., several thousand) over several kilometers for the above use cases, and may be at various distances away from the gateways or base stations (BSs). It thus becomes extremely challenging to connect and coordinate them for periodic or sporadic data collection and make time-critical, data-driven decisions.

To enable wide-area IoT and CPS applications, existing wireless sensor network (WSN) technologies (e.g., Zigbee and WirelessHART) form multi-hop mesh networks, complicating the protocol design and network deployment resulting in scalability issues, high energy consumption, and high latency

in data collection [1]. Due to their underlying design and operational limitations, existing low-power wide-area network (LPWAN) technologies (e.g., LoRa, SigFox, NB-IoT, and 5G) also suffer from scalability issues and high energy consumption and high latency in sensor data collection, especially in infrastructure-limited rural areas [2]. For example, the leading LPWAN technology, LoRa, supports approximately 120 nodes per 3.8 hectares until its performance drops sharply [3], which may not be sufficient to meet the scalability requirements of the emerging IoT and CPS applications [2].

In this paper, we focus on enabling massive scalability in an LPWAN technology called SNOW (sensor network over white spaces) [4]–[7]. The current SNOW design exploits the TV white spaces (i.e., allocated but locally unused TV channels and can be used by unlicensed devices [8]) to connect sensors to a BS. It has a D-OFDM (distributed orthogonal frequency-division multiplexing) based physical (PHY) layer that allows different *asynchronous* sensors to transmit data to a BS (i.e., uplink communication) concurrently using different D-OFDM subcarriers, where each sensor is assigned a subcarrier [4]. D-OFDM also allows a BS to transmit asynchronously to different sensors (i.e., downlink communication) concurrently using different subcarriers [5]. In this paper, we significantly advance the SNOW PHY layer where numerous sensors can *asynchronously* transmit to a BS simultaneously on the same subcarrier (when assigned the same subcarrier) while the rest of the subcarriers can be used in the similar fashion and in parallel by other sensors as well (uplink communication only).

Enabling massive concurrency in SNOW uplink is very challenging. **First**, concurrent transmissions from different sensors on the same D-OFDM subcarrier will *collide* and the BS may not decode any of these transmissions, resulting in lost packets and wasted energy consumption at the sensors. **Second**, parallel transmissions from different sensors on the neighboring D-OFDM subcarriers will break the *orthogonality* of the D-OFDM architecture, and hence the BS may not decode any of these transmissions, resulting in consequences similar to those above. In other words, these two reasons will introduce severe inter-symbol and inter-subcarrier interferences between signals from the sensors on the same subcarrier and neighboring subcarriers, respectively, resulting in decreased scalability in SNOW, high energy consumption at the sensors, and high latency at the BS in data collection in *convergecast* scenarios due to retransmissions.

In this paper, we address the above challenges as well as make the following key contributions:

- To enable successful concurrent transmissions from numerous asynchronous sensors to a BS on and across the D-OFDM subcarriers, we develop a set of decentralized *pseudorandom noise* (PN) sequences (also known as *pseudorandom spreading sequence*) based on Gold code [9]. Each sensor is assigned a PN sequence using which it asynchronously transmits and receives data on its subcarrier. Our set of PN sequences has very good cross-correlation properties (e.g., almost no correlation between the PN sequences), which minimizes the interference (e.g., inter-symbol interference and inter-subcarrier interference) on and across the D-OFDM subcarriers in SNOW.
- We enable a higher bitrate than the per-sensor bitrate requirement of the IEEE 802.15.4 standards' [10] direct-sequence spread spectrum (DSSS) where a group of 4 bits is spread to 32 chirps, considering a typical sensor data size of 28 bytes in practical deployments (e.g., those using TinyOS [11]). Our design may thus inspire enhanced scalability in the WSN standards as well.
- We develop a SNOW simulation platform using python's NumPy library and make it open-source [12]. To our knowledge, this is the first open-source simulation platform for SNOW. In our simulation, we implement the SNOW PHY layer, including our proposed innovations, and perform large-scale evaluation. The evaluation results show that our design provides approximately 9x improvement in scalability compared to the existing SNOW design, resulting in improved energy efficiency in the sensors and reduced latency at a BS in convergecast scenarios.

The rest of the paper is organized as follows. Section II presents the related work. Section III briefly overviews the existing SNOW architecture and presents our system model. Section IV details our PN sequences generation techniques for spreading and despreading data. Section V provides the implementation details and evaluation results. Finally, Section VI concludes our paper.

II. RELATED WORK

WSN Technologies. The emerging wide-area IoT and CPS applications need to connect and coordinate hundreds to thousands of sensors over distances of tens of kilometers. The existing WSN technologies operating in the 2.4 GHz spectrum (e.g., IEEE 802.15.4, IEEE 802.11, and BLE) may facilitate such connections by forming multi-hop mesh networks due to their short communication range [1], [13]. This, however, will complicate the protocol design, resulting in reduced scalability, high energy consumption at the sensors, high latency in data aggregation, and high cost in real-world deployments [6], [14]. In this paper, we develop protocols for enhanced scalability in LPWANs that have the potential to connect numerous sensors to a BS by forming a single-hop over several kilometers.

LoRa and Sigfox. Sigfox and LoRa are the two dominant LPWAN technologies operating in the *unlicensed ISM*

band [2]. Their devices adopt a 1% or 0.1% duty cycle requirement, making them less suitable for IoT or CPS applications with thousands of sensors or with real-time requirements [3], [15]–[19]. Sigfox supports a datarate of 10 to 1,000 bps, and a device can send at most 140 12-byte messages (each takes 3 seconds) per day. LoRa employs different channel bandwidths (BW) between 125 and 500 kHz, spreading factors (SFs) between 7 and 12, and coding rates between $\frac{4}{5}$ and $\frac{4}{8}$ to achieve scalability and different datarates. Using 125 kHz BW and SF of 10, a 12-byte payload in LoRa has an air time of 411.6 ms and bitrate of 980 bps. The higher the SF, the lower the bitrate in LoRa. This problem is exacerbated since large SFs are used more often [20]. Sigfox and LoRa may not be suitable for the emerging IoT and CPS applications requiring massive scale, high data rate, and ultra-low latency [2], [15]. Conversely, SNOW has the potential to achieve the above in the TV white spaces [7].

SNOW vs. Other LPWANs. A number of LPWAN technologies, including NB-IoT [21] and 5G [22] have targeted the cellular infrastructure and band. The 5G standard is currently under development. The NB-IoT specification froze at Release 13 of the 3GPP specification. Operating in the licensed band is costly due to high service fees and infrastructure and may not be available in the infrastructure-limited rural areas [2], [23], [24]. These technologies also require the sensors to frequently synchronize, which is much energy-consuming. It thus is impractical to ensure sustainability over an extended period, uninterrupted operation, and longevity of the emerging IoT and CPS applications. Many other technologies have been developed that operate in the licensed (e.g., LTE Cat M1 and EC-GSM-IoT) or unlicensed (e.g., INGENU, IQRF, Telensa, DASH7, Weightless-N/P, IEEE 802.11ah, IEEE 802.15.4k/g) bands [2], [15], [25]–[28] and severely interfere each other (as applicable). To avoid the high cost of the licensed band and the crowd of the ISM band, SNOW has been developed [4]–[7], [14], [29]–[34]. White spaces are widely available in both urban and rural areas, are less crowded, and offer a wider spectrum compared to other available frequencies for LPWANs [2], [5], [15], [35]. SNOW thus has huge potential, and we propose to significantly advance its PHY layer.

III. BACKGROUND AND SYSTEM MODEL

In this section, we briefly overview the SNOW technology and present our system model.

A. Overview of SNOW

Architecture. 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 (shown in Figure 1). The BS is powerful, Internet-connected, and line-powered while the nodes are power-constrained and do not access the Internet. To determine white space availability in a region, the BS queries a cloud-hosted geo-location database. A node depends on the BS to learn its

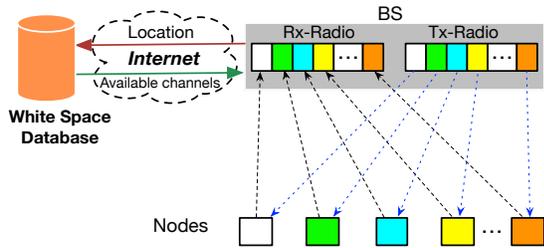


Fig. 1. SNOW Dual-radio BS and subcarriers.

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.

Physical Layer. 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 layer 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. The BS operates on a wideband channel split into overlapping (50%) orthogonal narrowband channels/subcarriers. Each node is assigned a subcarrier. For encoding and decoding on each subcarrier, the BS runs inverse fast Fourier transform (IFFT) and global fast Fourier transform (G-FFT) over the entire wideband channel, respectively. SNOW supports ASK (amplitude-shift-keying) and BPSK (binary phase-shift-keying) modulation techniques.

Medium Access Control Layer. When the number of nodes is no greater than the number of subcarriers, each node is assigned a unique subcarrier. Otherwise, a subcarrier is shared and the corresponding nodes use a lightweight CSMA/CA (carrier sense multiple access with collision avoidance)-based MAC (medium access control) protocol. The nodes can autonomously transmit, remain in receive mode, or sleep. 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 CCA (clear channel assessment). If the subcarrier is clear, 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).

B. System Model

Currently, in SNOW uplink, a BS can receive concurrently from distinct sensors using distinct subcarriers at any given instance (even with its MAC protocol), which limits the scalability. In our design, we proposed to enable concurrency within and across the subcarriers at any given instance in uplink communication, as depicted in Figure 2. Such concurrency will increase the scalability of SNOW by a factor of

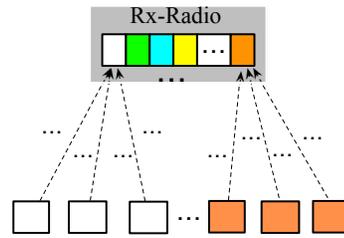


Fig. 2. Proposed concurrency in uplink for SNOW PHY layer.

$\sum_{i=0}^{r-1} s_i$, where s_i is the number of sensors using subcarrier i concurrently and r is the number of total subcarriers. In this paper, we limit our work to advancing the SNOW PHY layer in uplink only and leave the room for a new MAC protocol (needed when the number of nodes assigned to a subcarrier exceeds the subcarrier's concurrency capacity) and downlink communication as the future work. We thus solely focus on developing a set of PN sequences that preserves the *D-OFDM* feature of the SNOW PHY layer such that the inter-symbol interference and inter-subcarrier interference are minimal while the BS can decode data. Overall, the BS generates the set of PN sequences (and creates two instances of it) and assign each sensor a sequence when the sensor joins the network and assigned a subcarrier. No two sensors that are sharing a subcarrier or on neighboring subcarriers get the same PN sequence. We also adopt many design parameters of the current SNOW, including subcarrier overlapping (50%), bandwidth (200–400 kHz), and data modulation (ASK only).

IV. MASSIVE CONCURRENCY IN SNOW PHY LAYER

In this section, we detail our techniques for generating the set of PN sequences (i.e., pseudorandom spreading sequence), encoding (i.e., spreading) of the transmitted signals, and decoding (i.e., despreading) of the received signals. Additionally, we discuss the achievable datarates by our design.

A. Spreading Sequence Generation

Recall that we aim to enable concurrency within and across the *D-OFDM* subcarriers at any given instance. For this, we develop a set of PN sequences or waveforms, which allows numerous sensors to share a band of frequencies (i.e., subcarriers) with as little mutual interference as possible when each sensor is assigned a distinct sequence or code. Ideally, a received signal which has been spread using a different code will cause minimal interference in the aggregated signal over the entire bandwidth. The amount of interference from a sensor employing a distinct code (from a set) is related to the *cross-correlation* and power levels of all the codes in the set [36]. Unfortunately, such an ideal set would contain sequences of equally likely infinite random binary digits, requiring infinite storage in both the transmitter and receiver, which is impractical for the resource-constrained sensors.

The above limitations inspire the need for a set of *periodic* PN sequences (also used in Gold code [9]) that can be generated using a simplified circuit consisting of two linear

feedback shift registers (LFSRs) and a few XOR (exclusive OR) gates (one for XORing two LFSRs and one for each tap in the LFSRs), which is practical for the sensors. The number of taps in each LFSR is determined by its unique polynomial equation [37] and our achievable bitrate under minimum interference (explained in Section IV-D). An LFSR generates maximal-length sequences (m-sequences) that are the pseudorandom binary sequences of the maximum period (e.g., until it repeats). An XOR gate is used to mix two different m-sequences (of the same length) from two different LFSRs to generate a PN sequence in our design. In an LFSR, a bit is generated by a linear combination of the previous n bits, for a suitable choice of n . In a nutshell, a window of n bits (i.g., initial seed) is slide right (by one position) $2^n - 1$ times to cover $2^n - 1$ n -bit strings, generating $2^n - 1$ distinct m-sequences, each with a length of $2^n - 1$. We avoid 2^n slides since this starts repeating the sequences, which may cause inter-symbol interference within a subcarrier when the actual PN sequence is generated and used by the corresponding sensors. In the following, we detail the m-sequences and our intended set of PN sequences generation techniques.

m-sequences Generation. Each LFSR generates a maximum of $2^n - 1$ m-sequences, each of $2^n - 1$ bits as well, where n is the number of bits in the initial seed [38]. The register shifts all the bits to the right at each clock cycle c , generating the i -th sequence $a_i = (c_1 \odot a_{i-1}) \oplus (c_2 \odot a_{i-2}) \oplus \dots \oplus (c_n \odot a_{i-n})$ [39], which is a recursive formula. In the above equation, all the terms are binary (1 or 0), and \odot and \oplus are *modulo-2* multiplication and *modulo-2* addition operations, respectively. Specifically, the generated m-sequences with non-zero initial vectors (i.e., seeds) have period $N = 2^n - 1$ with the following three randomness properties that minimize the interference. (1) The number of 1's and 0's are approximately equal. (2) Half of the runs (i.e., subsequences of consecutive 1's and consecutive 0's) have length 1, $\frac{1}{4}$ runs have length 2, $\frac{1}{8}$ runs have length 3, and $\frac{1}{2^k}$ have length k , where ($k < n$). (3) It has sequence autocorrelation that is a randomness measurement and provides the degree of correspondence between an m-sequence and its phase-shifted replica. The smaller the correlation, the easier it is for a receiver to recover the m-sequence from interference. An m-sequence's periodic autocorrelation function R is given by $R(\tau) = \frac{1}{N} \sum_{n=1}^N a'_n a'_{n-\tau}$, where $a'_n = 1 - 2a_n$ (i.e., a ± 1 sequence) and τ represents different periods. It can also be shown that the periodic autocorrelation of an m-sequence is

$$R(\tau) = \begin{cases} 1 & \tau = 0, N, 2N, \dots \\ -\frac{1}{N} & \text{otherwise.} \end{cases}$$

Similar to autocorrelation, cross-correlation is also a measurement of interest in m-sequences. It is the degree of correspondence between m-sequences used by different users (i.e., sensors). Intuitively, the cross-correlation between different m-sequences needs to be low to avoid interference. If a'_n and b'_n are two m-sequences, then their cross-correlation $R_{a',b'}(\tau) = \frac{1}{N} \sum_{n=1}^N a'_n b'_{n-\tau}$, where $b'_n = 1 - 2b_n$ (i.e., a ± 1 sequence). It has been shown that the number of m-sequences

that have the least cross-correlation is very small and may not be feasible for multiple access systems [40], [41] such as D-OFDM due to asynchronicity between sensors within and across subcarriers. To this extent, we generate a set of PN sequences based on Gold code [39] using the generated m-sequences above.

Gold Code-Based PN Sequences Generation. Similar to the DS-CDMA (direct sequence code division multiple access) Gold codes, we generate a set of PN sequences for the D-OFDM system in SNOW such that different sensors may transmit asynchronously within and across subcarriers (which is unlike DS-CDMA). Gold codes provide a uniform and bounded cross-correlation between the codes [42]–[44]. Similar to the Gold code, our PN sequences are generated by repeatedly taking bitwise XORs of two *uncorrelated* m-sequences of the same length. Figure 3 shows such a generator. Two LFSRs with two non-zero seeds *seed 1* and *seed 2*,

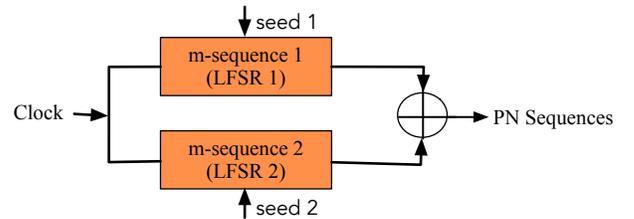


Fig. 3. Generation of PN sequences.

each of length n , generate two different m-sequences, each of length $N = 2^n - 1$. Changing the seeds generates a new set of PN sequences. For each PN sequence in a set, there may exist many pairs of m-sequences. However, not each pair of m-sequences generate a PN sequence that will have less cross-correlation (i.e., less mutual interference) with the other PN sequences in the set. For this, the PN sequences in D-OFDM should have three-valued peak cross-correlation magnitudes that are both uniform and bounded [40].

To generate a set of PN sequences with the above requirements, a good pair of m-sequences (also called *preferred pair*) is needed. Let our preferred pair be $\{u, v\}$ where u and v are generated by LFSR 1 and LFSR 2, respectively. If we consider u as a binary vector, then v can be generated in a deterministic manner by sampling every q -th bit of u , for some appropriate q (e.g., if and only if $\gcd(N, q) = 1$ [39]) from multiple copies of u until both u and v have the same length (e.g., N). The i -th PN sequence is generated by a bitwise XOR of u and an i -bit shifted copy of v . Specifically, $\{u, v\}$ should have the following properties. (1) Both LFSR 1 and LFSR 2 have preferred but unique polynomial equations with a degree of n . (2) n is not divisible by 4 [45]. (3) q is odd and either $q = (2^k + 1)$ or $q = (2^{2k} - 2^k + 1)$. (4) $\gcd(n, k) = 1$ if n is odd or $\gcd(n, k) = 2$ if n is even. Using the above technique, the set of PN sequences may be denoted as $G(u, v) = \{u, v, u \oplus v, u \oplus Dv, u \oplus D^2v, \dots, u \oplus D^{N-1}v\}$ where D is the delay element and represents the operator that shifts vectors cyclically to the left by one place. Additionally,

$G(u, v)$ contains a total of $M = (N + 2)$ sequences (recall that $N = (2^n - 1)$ and the "+2" is for initial referred pairs). In $G(u, v)$, any pair of PN sequences or a PN sequence and its shifted version has one of the three cross-correlation magnitudes in $\{-t(n), -1, t(n)-2\}$, where $t(n) = 1 + 2^{(n+2)/2}$ if n is even or $t(n) = 1 + 2^{(n+1)/2}$ if n is odd.

B. Encoding the Transmitted Signal

As discussed in Section III-B, we consider ASK, especially OOK (on-off keying), as the subcarrier modulation, where presence and absence of a carrier signal represent bit 1 and bit 0, respectively. Within a D-OFDM subcarrier, a sensor transmits a signal (which is a symbol in D-OFDM) to represent a data bit 1. To handle inter-subcarrier interference, a data bit is spread to 8 bits (actual symbol duration) by repeating it 8 times in the existing SNOW [6]. In our design, we spread a data bit to N bits by repeating it N times and then multiplying the spread data with the sensor's PN sequence (and subsequently mixed with the (sub)carrier signal). We thus have an N -bit long symbol that accounts for both inter-symbol interference (between the sensors within a subcarrier) and inter-subcarrier interference (between sensors across subcarriers). This way we achieve our proposed concurrency (as shown in Figure 2) in the SNOW PHY layer. Let $b_{ij}(k)$ and $g_{ij}(k)$ be the k -th spread bit of a data bit and k -th bit of the PN sequence of sensor j on subcarrier i , respectively. Thus, the signal for the k -th spread bit is $x_{ij}(k) = b_{ij}(k)g_{ij}(k)$. Overall, the symbol for a data bit 1 in our design may be represented as

$$[g_{ij}(k), g_{ij}(k + 1), \dots, g_{ij}(N - 1)]^T = g_{ij}.$$

We can create an equal-length (i.e., N -bit) symbol for data bit 0 with the similar process, which will be all 0's, and hence no signal transmission when mixed with the (sub)carrier signal.

C. Decoding the Received Signal

After the global FFT performed by the BS in SNOW (III-A), samples in each subcarrier are isolated (from the corresponding FFT bins) and considered for despreading and decoding by our system. Let, r_i be the received samples' vector of a symbol at subcarrier i . Each sample K in r_i is

$$r_i(K) = \sum_{i=1}^L x_{ij}(k) + z(K)$$

where L is the number of sensors using subcarrier i and z is the additive white Gaussian noise vector (AWGN). The power level (i.e., magnitude) of each sample is given by the G-FFT algorithm. Similar to the current SNOW PHY demodulator [4], we maintain a 2D matrix at the BS to decode each data bit from each sensor in each subcarrier. An entry $r_i(K)$ (interpreted as $r[i][K]$) in the matrix represents the K -th sample in i -th subcarrier. A decoding agent in the BS keeps running to detect data from different sensors on each subcarrier

by multiplying different PN sequences for that subcarrier. For example, the despread data from sensor j on subcarrier i is

$$r_i^T g_{ij} = [r_i(K), r_i(K+1), \dots, r_i(K+N-1)] \begin{bmatrix} g_{ij}(k) \\ \vdots \\ g_{ij}(k+N-1) \end{bmatrix}.$$

This operation gets rid of the interference (by other sensors if any) and noise. Note that the vectors of samples of symbols are generated right after the detection of a preamble of a packet in any subcarrier. After a symbol is despread, we recover the original data bit (which was repeated before spreading) by simply undoing the repeat operation. For this, we consider that a data bit is 1 if at least half of the repeated bits remain 1. This technique allows for an additional guard against interference.

D. Achievable Bitrate

For a sensor on an AWGN subcarrier of bandwidth B with signal-to-noise ratio (SNR) SNR , the maximum Tx bitrate $C = B \log_2(1 + SNR)$ based on the Shannon-Hartley Theorem. On a subcarrier with $B = 200$ kHz and $SNR = 3$ dB, we may achieve a bitrate of $\frac{200 \cdot 2}{N} = \frac{400}{N}$ kbps (recall that N is the PN sequence length of the sensor). In our evaluations (Section V), we choose $N = 7$, which gives us a Tx bitrate of ≈ 57.14 kbps per sensor. Additionally, two signal levels in our ASK modulation conform to the Nyquist Theorem $C = 2B \log_2 2^m$ where 2^m is the number of signal levels to support a theoretical bitrate of ≈ 57.14 kbps per sensor. If a subcarrier is shared by $M = (N + 2)$ number of sensors, then our maximum achievable Tx bitrate over bandwidth B increases M -times, which is $\approx M$ -times better (and conform to the Nyquist Theorem) compared to the IEEE 802.15.4 standards' datarate requirements of 50 kbps over a channel [10] or the current SNOW design. In evaluations, we, however, choose $B = 400$ kHz due to interference created by concurrent TxS and an $SNR = 6$ at the BS, which still provides us with an effective bitrate of > 50 kbps per sensor.

V. EVALUATION

We now present our implementation and evaluation results.

A. Implementation

We create a SNOW simulation platform using the Python programming language. For splitting a wideband into narrowband AWGN subcarriers, performing FFT operation, and other signal processing operations, we use the Python NumPy library. Similarly, we use the same library for generating the PN sequences and encoding of the transmitted signals at the sensors and decoding of the received signals at the BS. Our open-source implementation is available online [12].

B. Evaluation Setup

As discussed in our system model in Section III-B, no two sensors sharing the same subcarrier or are on neighboring subcarriers do not have the same PN sequence assigned to them. To ensure this, we generate two instances of our set of PN sequences (say, PN_{s_1} and PN_{s_2}) using two different sets

of initial seeds, which still hold the required cross-correlation properties. We then allocate PN_{s_1} to all the odd-numbered subcarriers and PN_{s_2} to all the even-numbered subcarriers, and thus ensuring the above requirements. For both PN_{s_1} and PN_{s_2} , we use $n = 3$, which yield $N = 7$, and thus aim for the discussed datarate in Section IV-D. For PN_{s_1} , we choose $seed_1 = seed_2 = 101$ (both can be the same since LFSRs use different polynomial equations) and get $PN_{s_1} = \{1011100, 1010011, 0001111, 1111011, 0010010, 1000001, 1100110, 0101000, 0110101\}$. For PN_{s_2} , we choose $seed_1 = seed_2 = 010$ (which is different from the seeds of PN_{s_1}) and get $PN_{s_2} = \{0101110, 0100111, 0001001, 1100000, 0110011, 0010100, 1011010, 1000111, 1111101\}$.

In simulation, we use 64 400kHz subcarriers (numbered 1 – 64) with 50% overlapping within 547 – 560 MHz (TV white spaces), and each subcarrier is shared by at most 9 sensors, totaling 576 sensors, which is 9-times higher than the original SNOW could accommodate. Similar to the current SNOW, we emulate a Tx power of 0 dBm, receive sensitivity of -85 dBm, packet size of 40 bytes (excluding an 1-byte preamble), containing 12-byte header and 28-byte random payload (data, CRC), and an SNR of 6 dB. Unless stated otherwise, these are our default parameter settings.

C. Threshold Selection

Since at most 9 sensors may transmit on the same subcarrier, the received signal strength (RSS) of the received symbols after the FFT output is not limited to 0s and 1s. Signals from concurrently transmitting sensors *superimpose* and make it challenging to decide the magnitude of the superimposed signal. For this, we consider the average signal power $\sum_{i=1}^M \sqrt{I^2 + Q^2}$ to decide different thresholds levels, where I and Q are the in-phase and quadrature signal components, and M is the averaging number of samples. Specifically, we collect 50,000 samples for each case when the number of concurrently transmitting sensors on a subcarrier vary from 0 to 9, where each sensor also transmits only 1s.

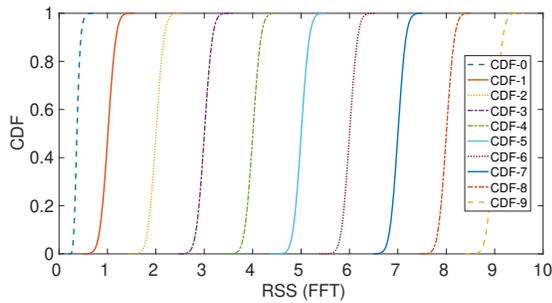


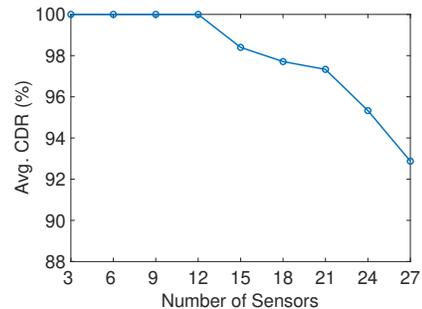
Fig. 4. Threshold behavior.

Figure 4 shows the *cumulative distribution function* (CDF) of RSS values in the above setup. As shown in this figure, when there is no transmission (i.e., no sensor), the RSS is below 0.5 for 100% of the cases, which may be used to denote magnitude 0. When a single sensor transmits, the RSS is between 0.51 and 1.5 for 100% of the cases, which may be

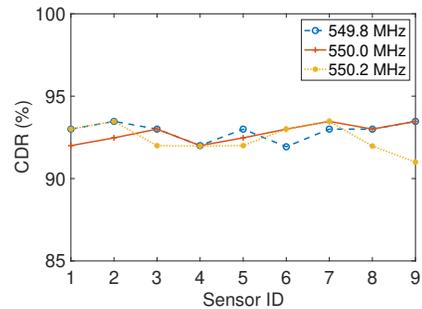
quantized to magnitude 1. For the case of 2 sensors, the RSS is between 1.51 and 2.5 for 100% of the cases, which may be quantized to magnitude 2. Similarly, to denote magnitudes 3, 4, 5, 6, 7, 8, and 9, the RSS ranges are 2.51 – 3.5, 3.51 – 4.5, 4.51 – 5.5, 5.51 – 6.5, 6.51 – 7.5, 7.51 – 8.5, and 8.51 – 9.5 for 100% of the cases, respectively. In the rest of the evaluations, we use the findings in this section to determine different magnitude levels, as necessary for despreading/decoding.

D. Evaluating Link Performance

In this section, we provide our evaluation results on reliability of the links (i.e., subcarriers) of our design. Specifically, we consider 3 neighboring subcarriers with 50% overlaps, which may generalize all the subcarriers in our implementation. For example, we use subcarriers with center frequencies 549.8 MHz, 550.0 MHz, and 550.2 MHz, where the subcarrier with 550.0 MHz center frequency is the middle subcarrier and overlaps 50% with its neighbors on both sides. For reliability calculation, we use the metric *correctly decoding rate* (CDR) that refers to the percentage of packets that are correctly decoded at the BS among all the transmitted ones by the sensors. For this simulation, we allow 1 to 9 sensors concurrently transmit on each of the considered subcarriers, totaling 27 sensors. In each case, a node sends consecutive 100 40-byte packets to the BS using its subcarrier with a random inter-packet interval of 0 – 3 ms that ensures overlapping of packets (each packet takes ≈ 5.6 ms to transmit) with other sensors on the same or neighboring subcarriers. We also repeat this experiment 100 times and present the results in Figure 5.



(a) Average correctly decoding rate



(b) CDR of subcarriers (extreme case).

Fig. 5. Link reliability in our design.

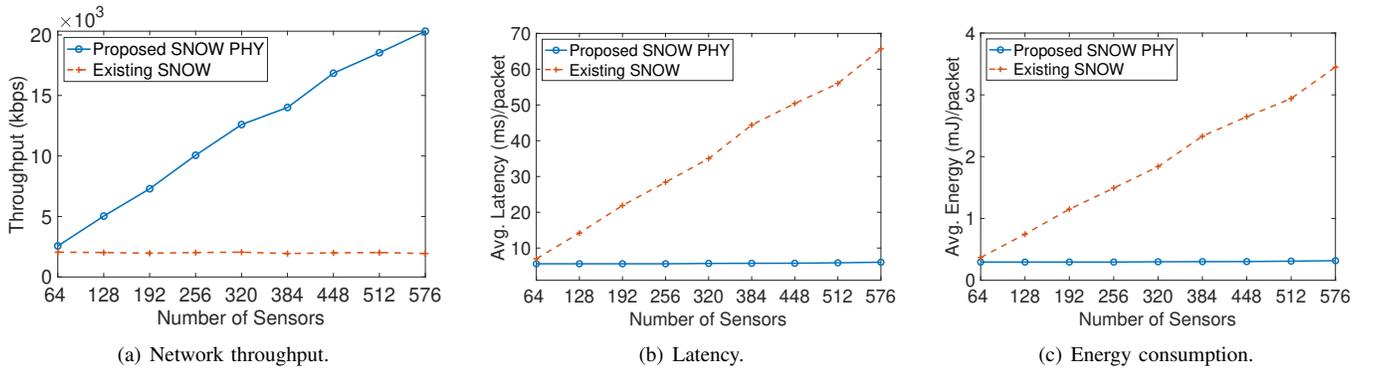


Fig. 6. Network performance evaluation.

Correctly Decoding Rate. Figure 5(a) shows the average CDR at the BS for the selected subcarriers for different number of sensors on different subcarriers. As shown in this figure, when 3, 6, 9, and 12 sensors transmit (i.e., 1, 2, 3, and 4 sensors on each subcarrier, respectively), the average CDRs of all these cases are 100%. For the cases where 15, 18, 21, 24, and 27 sensors transmit (i.e., 5, 6, 7, 8, and 9 on each subcarrier, respectively), the average CDRs are 98.4%, 97.71%, 97.33%, 95.33%, and 92.88%, respectively. In summary, in all the cases, the average CDRs are above 92%, which confirm high reliability of our design under massive concurrency and is acceptable in wireless networks [6], [14].

Subcarrier Reliability in Extreme Case. Figure 5(b) shows the CDRs on different subcarriers for the extreme case where 27 sensors transmit concurrently (i.e., 9 sensors on each subcarrier) and there are no inter-packet (of different sensors) delays. Thus, all the packets are colliding in the worst way possible in a network with concurrent transmissions in this scenario. As shown in Figure 5(b), for the packets (approximately 30,000 40-byte packets) of 1 sensor on each subcarrier (i.e., total 3 sensors), the CDRs on 549.8 MHz subcarrier is $\approx 93\%$, 550.0 MHz subcarrier is $\approx 92\%$, and 550.2 MHz subcarrier is $\approx 93\%$. This figure also shows that, as we increase the number of sensors on each subcarrier, the CDRs do not change drastically. For example, the extreme cases of 2, 3, 4, 5, 6, 7, 8, and 9 sensors on each subcarrier also yield CDRs in the approximate range of 92% – 93% in all the three selected subcarriers in this simulation setup.

E. Evaluating Network Performance

In this section, we evaluate several network parameters, including throughput (kbps), latency in data collection, energy consumption at the sensors, and performance under interference. Additionally, we compare our network performance with the existing SNOW design (MAC-enabled), as described in Section III-A. In this simulation, we use all the 64 subcarriers of the 547 – 560 MHz band. As mentioned earlier, all the sensors using odd- and even-numbered subcarriers get PN sequences from sets PN_{s_1} and PN_{s_2} , respectively. In this simulation, we create a convergecast scenario and analyze the following network parameters, where each sensor has 100 40-

byte packets (including 12-byte headers) with a random inter-packet interval of 0 – 3 ms.

Throughput. It refers to the effective bitrate at the BS. For calculating the throughput, we consider various numbers of sensors up to $9 \times 64 = 576$. Also, the 12-byte headers are not considered in the throughput calculation. Figure 6(a) shows the throughput of our design when numerous sensors between 64 and 576 transmit concurrently using 64 subcarriers (each having a minimum and maximum of 1 and 9 sensors, respectively). As shown in this figure, we achieve (vs. existing SNOW) a bitrate of approximately 2.56 Mbps (vs. 2.04 Mbps) and 5.03 Mbps (vs. 2.01 Mbps) when 64 and 128 sensors transmit concurrently. As we increase the number of sensors, our throughput increases almost linearly, unlike the fixed or slightly decreasing throughput in the existing SNOW (since it can decode only 64 sensors at any instance). For example, our bitrate is ≈ 20.31 Mbps (vs. ≈ 1.94 Mbps in existing SNOW) when 576 sensors transmit concurrently. We thus improve approximately 9x in throughput (vs. existing SNOW).

Latency. Figure 6(b) shows the per-packet latency in convergecast while taking into account the lost packets (without using ACK but by a curve fitting approach so that we can emulate a 100% reliability) as we increase from 64 to 576 sensors. Latency refers to the time it takes for a packet to be correctly delivered at the BS. As shown in this figure, the average (over sensors used) per-packet latency is approximately 5.6 ms when 64 sensors transmit concurrently in our design. As we increase the number of sensors up to 576, the average per-packet latency stays in the range 5.6 – 6.03 ms due to the massive concurrency in our design. In contrary, the average per-packet latency increases linearly or at a higher rate as we increase the number of sensors from 64 to 576 in the existing SNOW, which is due to its CSMA-based MAC. This simulation thus confirms timeliness in our design, which may help many time-critical or real-time IoT or CPS applications.

Energy Consumption. Figure 6(c) depicts the average (over sensors used) per-packet energy consumption (for 100% reliability) in our convergecast scenario. We calculate the energy based on the energy model of CC1310 transmitter (Tx current: 17.5 mA, idle current: 0.5 mA, and sleep current: $0.2\mu\text{A}$ at 0 dBm) that can operate in TV white spaces [4]. As shown

in Figure 6(c), our average per-packet energy consumption stays almost the same (in the approximate range of 0.2940 – 0.3166 mJ) when we transmit concurrently and increase total sensors from 64 to 576. The average per-packet energy consumption in the existing SNOW increases linearly or at a higher rate (in the approximate range of 0.3675 – 3.4493 mJ) as the total sensors is increased from 64 to 576. This is due to limited concurrency in SNOW and its CSMA-based MAC protocol. Our design thus shows better energy efficiency at the sensors, which will improve the lifetime of remote IoT or CPS applications. Overall, our simulations on several link parameters and network parameters show that our design provides much more concurrency in the PHY layer compared to the existing SNOW design.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we have significantly advanced the PHY layer of an LPWAN technology called SNOW by enabling unprecedented concurrency. In doing so, we have developed a set of PN sequences based on Gold code, which causes minimal interference within and across the SNOW subcarriers when used by the sensors. Our evaluation results have shown that we have achieved $\approx 9x$ more scalability in our design as well as significantly improved the per-packet latency and energy consumption at the network level. Overall, our design may motivate massive concurrency in communications in LPWANs and WSNs through its open-source implementation.

REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankar, and E. Cayirci, "Wireless sensor networks: a survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] D. Ismail *et al.*, "Low-power wide-area networks: opportunities, challenges, and directions," in *SCC*, 2018, pp. 1–6.
- [3] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do lora low-power wide-area networks scale?" in *MSWiM '16*, 2016, pp. 59–67.
- [4] A. Saifullah, M. Rahman *et al.*, "Snow: Sensor network over white spaces," in *SenSys '16*, 2016, pp. 272–285.
- [5] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra, "Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces," in *SenSys '17*, 2017, pp. 1–14.
- [6] —, "Low-power wide-area network over white spaces," *IEEE/ACM Trans. on Netw.*, vol. 26, no. 4, pp. 1893–1906, 2018.
- [7] M. Rahman, D. Ismail, V. P. Modekurthy, and A. Saifullah, "Implementation of lpwan over white spaces for practical deployment," in *IEEE/ACM IoTDI '19*, 2019, pp. 178–189.
- [8] "FCC, Second Memorandum Opinion and Order, ET Docket No FCC 10-174, September 2010," 2010. [Online]. Available: <https://fcc.gov>
- [9] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (corresp.)," *IEEE Trans. Inf. Theory*, vol. 13, no. 4, pp. 619–621, 1967.
- [10] "IEEE 802.11," 2022. [Online]. Available: <https://standards.ieee.org/ieee/802.15.4/7029/>
- [11] "TinyOS," 2022. <http://www.tinyos.net>.
- [12] "SNOW Simulation Platform," 2022. [Online]. Available: <https://github.com/lenslab8/snowSimulation>
- [13] S. Kim, S. Pakzad *et al.*, "Health monitoring of civil infrastructures using wireless sensor networks," in *IPSN '07*, 2007, pp. 254–263.
- [14] M. Rahman, D. Ismail, V. P. Modekurthy, and A. Saifullah, "Lpwan in the tv white spaces: A practical implementation and deployment experiences," *ACM TECS*, vol. 20, no. 4, pp. 1–26, 2021.
- [15] M. Rahman and A. Saifullah, "A comprehensive survey on networking over tv white spaces," *PMC*, vol. 59, pp. 1–17, 2019.
- [16] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in lora networks," in *EWSN '17*, 2017, pp. 323–328.
- [17] P. Marcelis, N. Kouvelas, V. S. Rao, and V. Prasad, "Dare: Data recovery through application layer coding for lorawan," *IEEE Trans. Mobile Comput.*, vol. 21, no. 3, pp. 895–910, 2020.
- [18] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of lora: Experiences from a large-scale measurement study," *ACM TOSN*, vol. 15, no. 2, pp. 1–35, 2019.
- [19] S. Fahmida, V. P. Modekurthy, M. Rahman, A. Saifullah, and M. Brocanelli, "Long-lived lora: Prolonging the lifetime of a lora network," in *ICNP '20*, 2020, pp. 1–12.
- [20] F. Adelantado *et al.*, "Understanding the limits of lorawan," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.
- [21] J. Chen *et al.*, "Narrowband internet of things: Implementations and applications," *IEEE IOT J.*, vol. 4, no. 6, pp. 2309–2314, 2017.
- [22] G. Akpakwu *et al.*, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.
- [23] D. Vasisht *et al.*, "Farmbeats: An iot platform for data-driven agriculture," in *NSDI '17*, 2017, pp. 515–529.
- [24] T. Chakraborty *et al.*, "Whisper: iotinthe tv white space spectrum," in *NSDI '22*, 2022, pp. 401–418.
- [25] N. Kouvelas, V. S. Rao, R. V. Prasad, G. Tawde, and K. Langendoen, "p-carma: Politely scaling lorawan," in *EWSN '20*, 2020, pp. 25–36.
- [26] P.-C. Hsieh *et al.*, "An experimental study on coverage enhancement of lte cat-m1 for machine-type communication," in *ICC '18*, 2018, pp. 1–5.
- [27] V. Saxena, A. Wallen, T. Tirronen, H. S. Razaghi, J. Bergman, and Y. Blankenship, "On the achievable coverage and uplink capacity of machine-type communications (mtc) in lte release 13," in *VTC-Fall '16*, 2016, pp. 1–6.
- [28] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Ham-moudeh, H. Gacanin, and U. M. Mbanaso, "Low-power wide area network technologies for internet-of-things: A comparative review," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2225–2240, 2018.
- [29] M. Rahman and A. Saifullah, "Integrating low-power wide-area networks for enhanced scalability and extended coverage," *IEEE/ACM Trans. on Netw.*, vol. 28, no. 1, pp. 413–426, 2020.
- [30] —, "Integrating low-power wide-area networks in white spaces," in *IEEE/ACM IoTDI '18*, 2018, pp. 255–260.
- [31] M. Rahman *et al.*, "Demo abstract: Enabling inter-snow concurrent p2p communications," in *IoTDI '18*, 2018, pp. 310–311.
- [32] D. Ismail *et al.*, "Demo abstract: Implementing snow on commercial off-the-shelf devices," in *IoTDI '18*, 2018, pp. 312–313.
- [33] M. Rahman, "Low-power wide-area network design," Ph.D. dissertation, Wayne State University, 2020.
- [34] V. P. Modekurthy, D. Ismail, M. Rahman, and A. Saifullah, "Low-latency in-band integration of multiple low-power wide-area networks," in *RTAS '21*, 2021, pp. 333–346.
- [35] P. Bahl *et al.*, "White space networking with wi-fi like connectivity," *ACM SIGCOMM '09*, vol. 39, no. 4, pp. 27–38, 2009.
- [36] Y. H. Kim, "System performance criteria in cdma networks using gold codes," Master's thesis, NJIT, 1988.
- [37] A. Klein, "Linear feedback shift registers," in *Stream Ciphers*. Springer, 2013, pp. 17–58.
- [38] S. Golomb, "Shift register sequences. aegean park press," 1982.
- [39] E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence cdma and wideband cdma cellular networks," *IEEE Commun. Mag.*, vol. 36, no. 9, pp. 48–54, 1998.
- [40] D. V. Sarwate and M. B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proc. IEEE*, vol. 68, no. 5, pp. 593–619, 1980.
- [41] Y.-H. Lee and S.-J. Kim, "Sequence acquisition of ds-cdma systems employing gold sequences," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2397–2404, 2000.
- [42] P. Fan, M. Darnell, and M. Darnell, *Sequence design for communications applications*. Research Studies Press Ltd, 1996, vol. 1.
- [43] S. Pradhan *et al.*, "A novel orthogonal minimum correlation spreading code in cdma system," *IJWMT*, vol. 2, pp. 38–52, 2014.
- [44] R. Gold, "Maximal recursive sequences with 3-valued recursive cross-correlation functions (corresp.)," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 154–156, 1968.
- [45] S. W. Golomb, "Shift-register sequences and spread-spectrum communications," in *ISSSTA '94*. IEEE, 1994, pp. 14–15.