Low-Power Wide-Area Network Over White Spaces

Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, *Fellow, IEEE*, Jie Liu, *Fellow, IEEE*, and Ranveer Chandra

***Abstract*— As a key technology driving the Internet-of-Things, low-power wide-area networks (LPWANs) are evolving to overcome the range limits and scalability challenges in traditional wireless sensor networks. This paper proposes a new LPWAN architecture called *sensor network over white spaces (SNOW)* by exploiting the TV white spaces. The SNOW is the first highly scalable LPWAN over TV white spaces that enable asynchronous, bi-directional, and massively concurrent communication between numerous sensors and a base station. This is achieved through a set of novel techniques. The SNOW has a new OFDM-based physical layer that allows the base station using a single antenna-radio: 1) to send different data to different nodes concurrently and 2) to receive concurrent transmissions made by the sensor nodes asynchronously. It has a lightweight media access control protocol that: 1) efficiently implements per-transmission acknowledgments of the asynchronous transmissions by exploiting the adopted OFDM design and 2) combines CSMA/CA and location-aware spectrum allocation for mitigating hidden terminal effects, thus enhancing the flexibility of the nodes in transmitting asynchronously. We implement the SNOW in GNU radio using universal software radio peripheral devices. Experiments through deployments in three radio environments—a large metropolitan city, a rural area, and an indoor environment—as well as large-scale simulations demonstrated that the SNOW drastically enhances the scalability of a sensor network and outperforms existing techniques in terms of scalability, energy, and latency.**

***Index Terms*— White space, sensor network, LPWAN, OFDM.**

I. Introduction

TODAY, wireless sensor networks (WSNs) are emerging in large-scale and wide-area applications (e.g., urban sensing [1], oil field management [2], and precision agriculture [3]) that often need to connect thousands of sensors over long distances. Existing WSN technologies operating in the ISM bands such as IEEE 802.15.4 [4], IEEE 802.11 [5], and Bluetooth [6] have short range (e.g., 30-40m for IEEE 802.15.4 in 2.4GHz) that poses a significant challenge in meeting this impending demand. To cover a large area with numerous devices, they form multi-hop mesh networks at the expense of energy, cost, and complexity, limiting scalability. Low-Power Wide-Area Network (LPWAN) is becoming a promising Internet of Things (IoT) technology to overcome these range limits and scalability challenges [7], [8]. In this paper, we propose a highly scalable LPWAN architecture called ***Sensor Network Over White Spaces (SNOW)*** by designing sensor networks to operate over TV white spaces. *White spaces* refer to the allocated but locally unused TV channels, and can be used by unlicensed devices [9]. Compared to existing LPWAN technologies, SNOW offers higher scalability and energy-efficiency and takes the advantages of free TV white spaces.

Compared to the ISM bands, white spaces offer a large number of, less crowded channels, each 6MHz, in both rural and urban areas [10], [11]. The Federal Communications Commission (FCC) in the US mandates that a device needs to either sense the channel before transmitting, or consult with a cloud-hosted geo-location database [9] to determine the white spaces at a location. Similar regulations are adopted in many countries. Thanks to their lower frequencies (54 – 862MHz in the US), white spaces have excellent propagation characteristics over long distance and through obstacles, and hence hold enormous potential for WSN applications that need long communication range. To date, this potential has been exploited mostly for broadband access by industry leaders such as Microsoft [12] and Google [13] as well as by various standards bodies such as IEEE 802.11af [14], IEEE 802.22 [15], and IEEE 802.19 [16]. In contrast, our objective is to exploit them for wide-area, large-scale WSNs. Long transmission range will reduce many WSNs to single-hop that has potential to avoid the complexity, overhead, and latency associated with multi-hop. Such a paradigm shift faces the challenges that stem from long range such as increased chances of packet collision. It must also satisfy the typical requirements of WSNs such as low cost nodes, scalability, reliability, and energy efficiency.

We address the above challenges and requirements of WSN in the SNOW design. SNOW is the first design of a highly scalable low power and long range WSN over the TV white spaces initially presented in [17] and [18]. At the heart of its design is a Distributed implementation of Orthogonal Frequency Division Multiplexing (OFDM), called ***D-OFDM***. The **base station (BS)** splits the wide white space spectrum into narrowband orthogonal subcarriers allowing D-OFDM to carry parallel data streams to/from the distributed nodes from/to the BS. Each sensor uses only one narrow-band radio. The BS uses two wide-band radios, one for transmission and the other for reception, allowing transmission and reception in parallel. Each radio of the BS and a sensor is half-duplex and equipped

Manuscript received August 22, 2017; revised February 13, 2018 and April 12, 2018; accepted June 19, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Chen. Date of publication August 7, 2018; date of current version August 16, 2018. This work was supported by the NSF under Grants CRII-1565751 (NeTS), CNS-1320921 (NeTS), and 1646579 (CPS), and in part by the Fullgraf Foundation. *(Mahbubur Rahman and Abusayeed Saifullah are co-first authors.) (Corresponding author: Abusayeed Saifullah.)*

A. Saifullah, M. Rahman, and D. Ismail are with the Department of Computer Science, Wayne State University, Detroit, MI 48202 USA (e-mail: saifullah@wayne.edu).

C. Lu is with the Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA.

J. Liu and R. Chandra are with Microsoft Research, Redmond, WA 98052 USA.

Digital Object Identifier 10.1109/TNET.2018.2856197

with a single antenna. SNOW supports reliable, concurrent, and asynchronous receptions with one single-antenna radio and multiple concurrent data transmissions with the other single-antenna radio. This is achieved through a new physical layer (PHY) design by adopting D-OFDM for multiple access in both directions and through a lightweight Media Access Control (MAC) protocol. While OFDM has been embraced for multiple access in various wireless broadband and cellular technologies recently, its adoption in low power, low data rate, narrowband, and WSN design is novel. Taking the advantage of low data rate and short payloads, we adopt OFDM in WSN through a much simpler and energy-efficient design.

The key contributions of this paper are as follows.

- We design a D-OFDM based PHY for SNOW with the following features for scalability, low power, and long range. **(1)** Using a single-antenna radio, the BS can receive concurrent transmissions made by the sensor nodes asynchronously. **(2)** Using a single-antenna radio, the BS can send different data to different nodes concurrently. **(3)** The BS can exploit fragmented white space spectrum. Note that the above design is different from MIMO radio adopted in various wireless domains such as LTE, WiMAX, and IEEE 802.11n [19] as they rely on multiple antennas to enable multiple transmissions and receptions. Setting up multiple antennas is expensive and **difficult** for lower frequencies due to large form factor and required space (half of wavelength) between antennas.
- We develop a lightweight MAC protocol that handles subcarrier allocation and operates the nodes with flexibility, low power, and reliability. It has the following features. **(1)** Considering a single half-duplex radio at each node and two half-duplex radios at the BS, we efficiently implement per-transmission ACK of the asynchronous transmissions by taking the advantage of D-OFDM design. **(2)** It combines CSMA/CA and location-aware subcarrier assignment for mitigating hidden terminals effects, thus enhancing the flexibility of the nodes that need to transmit asynchronously. **(3)** The other features include the capability of handling peer-to-peer communication, load balancing, and spectrum and network dynamics.
- We implement SNOW in GNU Radio [20] using Universal Software Radio Peripheral (USRP) [21] devices. In our experiments, a single radio of the SNOW BS can encode/decode 29 packets on/from 29 subcarriers within 0.1ms to transmit/receive simultaneously, which is similar to standard encoding/decoding time for one packet.
- We perform experiments through SNOW deployments in three different radio environments - a city, a rural area, an indoor testbed. Both experiments and large-scale simulations show its high efficiency in terms of latency and energy with a linear increase in throughput with the number of nodes, demonstrating its superiority in scalability over existing designs.

Organization: Sections II, III, and IV describe the network architecture, PHY, and MAC protocol of SNOW, respectively.

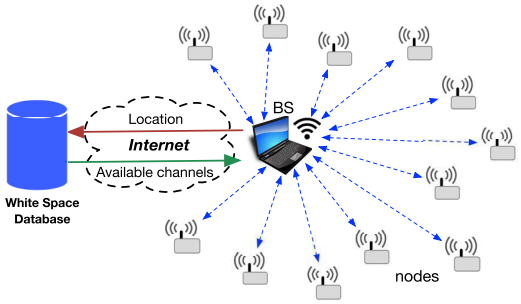


Fig. 1. SNOW architecture.

Sections V and VI present implementation and experiments, respectively. Sections VII and VIII compare SNOW with other LPWANs and related work. Section IX is the conclusion.

II. SNOW Architecture

A WSN is characterized by small packets, low data rate, and low power [4], [22]. The nodes are typically battery-powered. Thus, scalability and energy efficiency are the key concerns in WSN design. We consider a lot of sensor nodes associated with a BS. Each sensor node (called **'node'** throughout the paper) is equipped with a single half-duplex narrow-band radio operating in white space. Due to long transmission (Tx) range even at low power (e.g., several kilometers at 0dBm in our experiment in Section VI) of this radio, we consider that the nodes are directly connected (with a single hop) to the BS and vice versa as shown in Figure 1. However, the nodes may or may not be in communication ranges of the other nodes. That is, some nodes can remain as hidden terminal to some other nodes. The BS and its associated nodes thus form a star topology. The nodes are power constrained and not directly connected to the Internet.

The BS uses a wide spectrum as a single channel that is split into subcarriers, each of equal spectrum width (bandwidth). Each node is assigned one subcarrier on which it transmits to and receives. For integrity check, the senders add cyclic redundancy check (CRC) at the end of each packet. We leave most complexities at the BS and keep the nodes simple and energy-efficient. The nodes do not do spectrum sensing or cloud access. The BS retrieves white space channels by inputting the locations of its own and all other nodes into a cloud-hosted database through the Internet as shown in Figure 1. We assume that it knows the locations of the nodes through manual configuration or some existing WSN localization technique such as those based on ultrasonic sensors or other sensing modalities [23]. Localization is out of the scope of this paper. We use two radios at the BS to support concurrent transmission and reception which will be described in Section IV.

III. SNOW PHY Layer Design

The PHY layer of SNOW is designed to achieve scalable and robust bidirectional communication between the BS and numerous nodes. Specifically, it has three key design goals: (1) to allow the BS to receive concurrent and asynchronous transmissions from multiple nodes using a single antenna-radio; (2) to allow the BS to send different packets to multiple nodes concurrently using a single antenna-radio; and (3) to allow the BS to exploit fragmented spectrum.

A. Design Rationale

To achieve scalability and energy efficiency, the PHY layer of SNOW is designed using a **D**istributed implementation of **OFDM** for multi-user access, called **D-OFDM** in this paper. *OFDM* is a frequency-division multiplexing scheme to carry data on multiple parallel streams between a sender and a receiver using many orthogonal subcarrier signals. It has been adopted for multi-access in the forms of OFDMA (Orthogonal Frequency Division Multiple Access) and SC-FDMA (Single Carrier Frequency Division Multiple Access) in some broadband and cellular technologies recently [24] both of which require strong time synchronization among nodes. As a major difference from those, D-OFDM enables multiple receptions using a single antenna and also enables different data transmissions to different nodes using a single antenna, and does not need time synchronization.

In SNOW, the BS's wide white space spectrum is split into narrowband orthogonal subcarriers which carry parallel data streams to/from the distributed nodes from/to the BS as D-OFDM. Narrower bands have lower bit rate but longer range, and consume less power [25]. We adopt D-OFDM by assigning the orthogonal subcarriers to different nodes. Each node transmits and receives on the assigned subcarrier. Each subcarrier is modulated using Binary Phase Shift Keying (BPSK) which is highly robust due to a difference of $180°$ between two constellation points.

The key feature in OFDM is to maintain subcarrier orthogonality. If the integral of the product of two signals is zero over a time period T, they are ***orthogonal*** to each other. Two sinusoids with frequencies that are integer multiples of a common one satisfy this criterion [26], i.e., two subcarriers at center frequencies f_i and f_j are orthogonal when over T:

$$\int_0^T \cos(2\pi f_i t)\cos(2\pi f_j t)dt = 0.$$

The orthogonal subcarriers can be **overlapping**, thus increasing spectral efficiency. As long as orthogonality is maintained, it is possible to recover the individual subcarriers' signals. In the **downward communication** in SNOW (when a single radio of the BS transmits different data to different nodes using a single transmission), OFDM encoding happens at a single radio at the BS while the distributed nodes decode their respective data from their respective subcarriers. In the **upward communication** (when many nodes transmit on different subcarriers to the BS), OFDM encoding happens in a distributed fashion on the nodes while a single radio at the BS decodes their data from the respective subcarriers.

Let the BS spectrum is split into n orthogonal subcarriers: $f_1, f_2, f_3, \cdots, f_n$. Then, it can receive from at most n nodes simultaneously. Similarly, it can carry n different data at a time. When the number of nodes is larger than n, a subcarrier is shared by multiple nodes and their communication is governed by the MAC protocol (Section IV). To explain the PHY design we ignore subcarrier allocation and consider only the n nodes that have occupied the subcarriers for transmission.

B. Upward Communication

Here we describe how we enable parallel receptions at a single radio at the BS. In D-OFDM, we adopt Fast Fourier Transformation (FFT) to extract information from all subcarriers. We allow the nodes to transmit on their respective subcarriers whenever they want without coordinating among themselves. Figure 2 shows a workflow of the steps for decoding packets from multiple subcarriers at the BS.

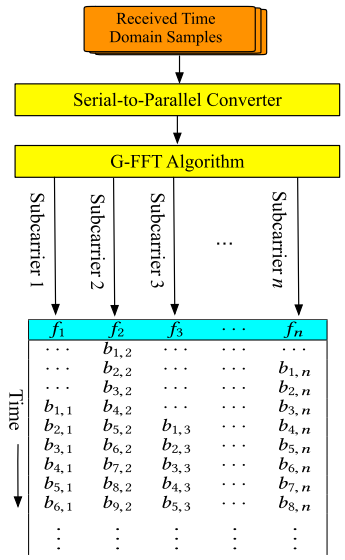


Fig. 2. The steps of a packet decoding.

Every node independently encodes based on BPSK the data on its subcarrier. To decode a composite OFDM signal generated from orthogonal subcarriers from the distributed nodes, we adopt FFT as a ***Global FFT Algorithm (G-FFT)*** which runs a single FFT algorithm on the entire BS spectrum, instead of running a separate FFT to decode each of the concurrently received packets. Specifically, G-FFT runs a single FFT algorithm even if the BS spectrum is not continuous (i.e. some parts of the spectrum is unavailable or unused). Such an approach will help us decode asynchronous transmissions and also exploit fragmented white space spectrum using a single radio with a single FFT module. To receive asynchronous transmissions, the BS keeps running the G-FFT algorithm. Every incoming packet on any subcarrier follows preamble bits for packet detection. Once a preamble is detected on a subcarrier, the receiver immediately gets ready to receive subsequent bits of the packet. A vector v of size equal to the number of FFT bins stores the received time domain samples. G-FFT is performed on v at every cycle of the baseband signal. For n subcarriers, we apply an m point G-FFT, where $m \geq n$ (m is a multiple of n). Each subcarrier corresponds to $\frac{m}{n}$ bins with one middle bin representing its center frequency. The frequency bins are ordered from left to right with the left most $\frac{m}{n}$ bins representing the first subcarrier (f_1). Each FFT output gives a set of m values. Each index in that set represents a single energy level and phase of the transmitted sample at the corresponding frequency at a time instant.

In BPSK, bit 0 and 1 are represented by keeping the phase of the carrier signal at $180°$ and $0°$ degree, respectively. We use a phase threshold that represents maximum allowable phase deviation in the received samples. One symbol is mapped into one bit. Since any node can transmit any time without any synchronization, the decoding of all packets is handled

by maintaining a 2D matrix where each column represents a subcarrier or its center frequency bin that stores the bits decoded at that subcarrier. The last step in Figure 2 shows the 2D matrix where entry $b_{i,j}$ represents the i-th bit (for BPSK) of subcarrier f_j. The same process thus repeats.

1) Handling Spectrum Leakage: The G-FFT algorithm works on a finite set of time domain samples that represent one period of the signal. The captured signal may not be an integer multiple of periods, resulting in a truncated waveform. Thus, FFT outputs some spectral components that are not in the original signal, letting the energy at one spectral component leak into the neighboring ones. To mitigate the effects of such *spectral leakage*, we adopt *Blackman-Harris windowing* [27]. *Windowing* multiplies a discontinuous time domain record by a finite length window. This window works for random or mixed signals and has amplitudes that vary smoothly and gradually towards zero at the edges, minimizing the effects of leakage.

2) Handling Carrier Frequency Offset: In OFDM communication, the orthogonal subcarriers are subject to carrier frequency offset (CFO), thereby loosing orthogonality and introducing inter-carrier interference (ICI). CFO stems from a frequency mismatch between the local oscillators at the transmitter and receiver due to hardware non-ideality and also from the Doppler shift, which is a result of the relative motion between the transmitter and receiver in mobile environments. ICI caused by CFO attenuates the desired signal reducing the signal-to-noise ratio. Hence, for enhanced performance of an OFDM system, CFO needs to be estimated and compensated for.

We use training symbols (preamble) for CFO estimation. Due to distributed and asynchronous nature of SNOW, CFO estimation in D-OFDM is done in a slightly different way compared to traditional OFDM. To estimate in absence of ICI, CFO estimation in D-OFDM is done when a node joins the network. For node joining, SNOW uses one (or more) subcarrier, called *join subcarrier*, that does not overlap with any other subcarrier. Each node joins the network by first communicating with the BS on a join subcarrier. Each way communication follows preamble that is used to estimate CFO on join subcarrier. Specifically, preamble from a node to BS allows to estimate CFO at the BS, and that from BS to a node allows to estimate CFO at the node on the join subcarrier. Later, based on the CFO on a join subcarrier, we determine the CFO on a node's assigned subcarrier as described below. CFO estimation technique for both upward and downward communication is similar. However, we adopt different approach for CFO compensation in upward and downward communication. We first describe the CFO estimation technique.

First we explain how we estimate CFO on a join subcarrier f. Since it does not overlap with other subcarriers, it is ICI-free. If f_{Tx} and f_{Rx} are the frequencies at the transmitter and at the receiver, respectively, then their frequency offset $\Delta f = f_{Tx} - f_{Rx}$. For transmitted signal $x(t)$, the received signal $y(t)$ that experiences a CFO of Δf is given by

$$y(t) = x(t)e^{j2\pi\Delta f t} \tag{1}$$

We estimate Δf based on short and long preamble approach, similar to IEEE 802.11a [28], using time-domain samples. In our implementation, we divide a 32-bit preamble into two equal parts, each of 16 bits. First part is for coarse estimation and the second part is for finer estimation of CFO [28]. Considering δt as the short preamble duration,

$$y(t-\delta t) = x(t)e^{j2\pi\Delta f(t-\delta t)}.$$

Since $y(t)$ and $y(t-\delta t)$ are known at the receiver,

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

Taking angle of both sides,

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

Thus,

$$\Delta f = -\frac{\sphericalangle y(t-\delta t)y^*(t)}{2\pi\delta t}$$

A SNOW node calculates the CFO on join subcarrier f using the preambles from the BS to the node using the above approach. Note that, for upward communication, the BS keeps running G-FFT on the entire BS spectrum including the join subcarrier as other nodes may be transmitting to it. Therefore, the G-FFT outputs for the join subcarrier are converted to time-domain samples using Inverse FFT (IFFT). These time-domain samples are used for CFO estimation on the join subcarrier f at the BS based on the above approach. Then the *ppm (parts per million)* on the receiver's (BS or SNOW node) crystal is given by $\text{ppm} = 10^6\frac{\Delta f}{f}$. Thus, the receiver (BS or a node) calculates Δf_i on subcarrier f_i as

$$\Delta f_i = \frac{f_i * \text{ppm}}{10^6}.$$

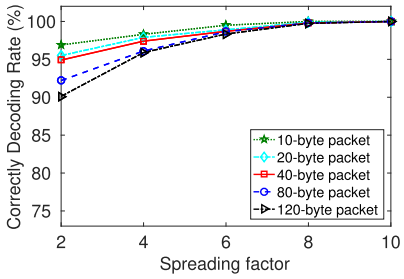
Thus the BS and a SNOW node that is assigned subcarrier f_i calculates CFO on f_i on its respective side. To take into account Doppler shift, CFO has to be estimated using the above approach while a node moves. For simplicity, we do not consider mobility and ignore CFO due to Doppler shift.

As the nodes asynchronously transmit to the BS, doing the CFO compensation for each subcarrier at the BS is quite difficult. Hence we adopt a simple feedback approach for proactive CFO correction in upward communication. Δf_i estimated at the BS for subcarrier f_i is given to the node (through downward transmission) that is assigned f_i during its joining process. The node can then adjust its transmitted signal based on Δf_i (when transmitting on subcarrier f_i) which will align its signal so that the BS does not need to compensate for Δf_i. Such feedback based proactive compensation scheme was studied before for multiple access OFDM [29] and is also used in global system for mobile communication (GSM).

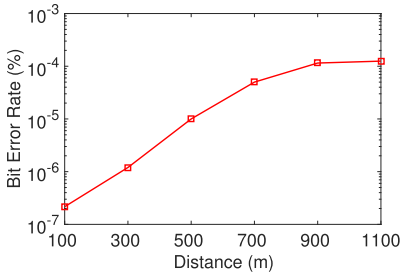
C. Downward Communication

One of our key objectives is to enable transmission from the BS which will encode different data on different subcarriers. A node's data will be encoded on the associated subcarrier. The BS then makes a single transmission and all nodes will decode data from their respective subcarriers. In the following, we describe our technique to achieve this.

Our goal in D-OFDM is to enable distributed demodulation at the nodes without any coordination among them.



(a)



(b)

Fig. 3. Determining spreading factor. (a) CDR under varying SF, packet sizes. (b) BER over distances when SF = 8.

That is, from the received OFDM signal, every node will independently decode the data from the signal component on its subcarrier only. The main design technique lies in the encoding part at the BS. We enable this by adopting IFFT (Inverse FFT) at the transmitter side. IFFT is performed after encoding data on the subcarriers. We can encode data on any subset of the subcarriers. The transmission is made after IFFT. If the OFDM transmitter uses m point IFFT algorithm, consecutive m symbols of the original data are encoded in m different frequencies of the time domain signal with each run of IFFT. We encode different symbols for different nodes on different subcarriers, thus obviating any synchronization between symbols. We use a vector v of size equal to the number of IFFT bins. Each index of v is a frequency bin. If the BS has any data for node i, it maps one unit of the data to a symbol and puts in the i-th index. If it has data for multiple nodes, it creates multiple symbols and puts in the respective indices of v. Then the IFFT algorithm is performed on v and a composite time domain signal with data encoded in different frequencies is generated and transmitted. This repeats at every cycle of baseband signal. A node listens to its subcarrier center frequency and receives only the signal component in its subcarrier frequency. The node then decodes data from it.

Handling CFO. In Section III-B, we have already described how a node that is assigned subcarrier f_i estimates CFO Δf_i. In downward communication, the node compensates for CFO in time-domain using Equation (1).

D. Using Fragmented Spectrum

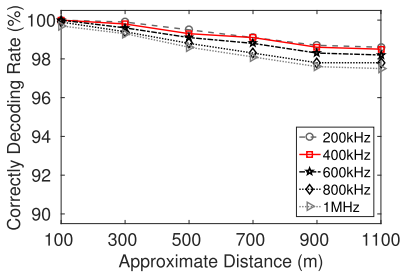
An added advantage of our design is that it allows to use fragmented spectrum. When we cannot find consecutive white space channels while needing more, we may use non-consecutive channels. The G-FFT and IFFT algorithms will be run on the entire spectrum as a single wide channel that includes all fragments (and the occupied TV channels between them). The occupied spectrum will not be assigned to any node and the corresponding bins will be ignored in decoding and encoding in G-FFT and IFFT, respectively.

E. Design Considerations

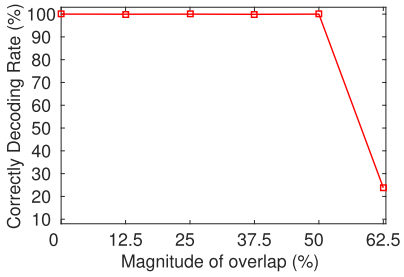
1) Link Parameters: Bit spreading is a technique to reduce bit errors by transmitting redundant bits for ease of decoding in noisy environments [4], [5]. By adopting a proper spreading factor, its effects can be made similar to extended Cyclic Prefix (CP), thereby significantly mitigating inter-symbol interference (ISI). Specifically, in D-OFDM, time synchronization is avoided by extending the symbol duration (repeating a symbol multiple times) and sacrificing bit rate. The effect is similar to extending CP beyond what is required to control ISI. CPs of adequate lengths have the effect of rendering asynchronous signals to appear orthogonal at the receiver, increasing the guard-interval. As it reduces data rate, D-OFDM is suitable for LPWANs. Using USRP devices in TV white spaces and using narrow bandwidth (400kHz) we tested with different packet sizes and bit spreadings factor (SF). We define ***Correctly Decoding Rate (CDR)*** - as the ratio of the number of correctly decoded packets at the receiver to the total number of packets transmitted. A receiver can always decode over 90% of the packets when the sender is 1.1km away and transmits at 0 dBm (Figure 3(a)). Figure 3(b) shows that bit error rate (BER) remains negligible under varying distances (tested up to 1.1km). For wireless communications, a packet is usually dropped if its BER exceeds 10^{-3} [30]. Thus we will use SF=8 as our experiments found it to be sufficient for robust communication. We have tested the feasibility of different packet sizes (Figure 3(a)). WSN packet sizes are usually short. For example, TinyOS [31] (a platform/OS for WSN motes based on IEEE 802.15.4) has a default payload size of 28 bytes. We use 40-byte (28 bytes payload + 12 bytes header) as our default packet size in our experiment.

Note that, like many other LPWANs (e.g., LoRa, SigFox) and most WSN devices, we also do not do channel estimation to keep node design very simple. Choosing an effective bit spreading factor allows us to decode without estimating channel. It is understandable that channel state information can help us better mitigate the multipath effects, specially in indoor environments. In the future, we shall study the trade-offs between the overhead of channel estimation in low-power node design and the reliability gain through it.

2) Subcarriers: The maximum transmission bit rate R of an AWGN channel of bandwidth W' based on Shannon-Hartley Theorem is given by $R = W' \log_2(1 + SNR)$, where *SNR* is the *Signal to Noise Ratio*. Based on Nyquist Theorem, $R = 2W' \log_2 2^k$ where k is the number of bits per symbol (2^k being the number of signal levels) needed to support bit rate R for a noiseless channel. The 802.15.4 specification for lower frequency band, e.g., 430-434MHz band (IEEE 802.15.4c [32]), has a bit rate of 50kbps. We also aim to achieve this bit rate. We consider a minimum value of 3dB for SNR in decoding. Taking into account default $SF = 8$, we need to have $50 * 8$kbps bit rate in the medium. Thus,



(a)



(b)

Fig. 4. Determining subcarriers. (a) Reliability over distance. (b) Overlaps between subcarriers.

a subcarrier of bandwidth 200kHz can have a bit rate up to $50 * 8$kbps in the medium. Since BPSK has $k = 1$, it is theoretically sufficient for this bit rate and bandwidth under no noise. Using similar setup as the above, Figure 4(a) shows the feasibility of various bandwidths. In our experiments, 400kHz bandwidth provides our required bit rate under noise. Hence, we use 400kHz as our default subcarrier bandwidth. We have also experimentally found that our 400kHz subcarriers can safely overlap up to 50% with the neighboring ones (as shown in Figure 4(b)). In our low data rate communication, using a spreading factor of 8 helps us mitigate ISI.

IV. SNOW MAC Protocol

We develop a lightweight MAC protocol for operating the nodes with flexibility, low power, and reliability. As the nodes transmit asynchronously to the BS, implementing ACK for every transmission is difficult. Considering a single half-duplex radio at each node and two half-duplex radios (both operating on the same spectrum) at the BS, we demonstrate that we can implement ACK immediately after a transmission. Under such a design decision in SNOW, we can exploit the characteristics of our D-OFDM system to enable concurrent transmissions and receptions at the BS.

A. Location-Aware Spectrum Allocation

This BS spectrum is split into n overlapping orthogonal subcarriers: $f_1, f_2, \cdots, f_n$, each of equal width. Considering w as the subcarrier bandwidth, W as the total bandwidth at the BS, and α as the magnitude of overlap of the subcarriers (i.e., how much two neighboring subcarriers can overlap),

$$n = \frac{W}{w\alpha} - 1.$$

For example, when $\alpha = 50\%$, $W = 6$MHz, $w = 400$kHz, we can have $n = 29$ orthogonal subcarriers. The BS can use a vector to maintain the status of these subcarriers by keeping their noise level or airtime utilization (considering their usage by surrounding networks), and can dynamically occupy or leave some subcarrier. Since our PHY can use fragmented spectrum, such dynamism at the MAC layer is feasible.

Subcarrier allocation is done at the BS. Each node is assigned one subcarrier. Let $g(u)$ denote the subcarrier assigned to node u. When the number of nodes is no greater than the number of subcarriers, i.e. $N \leq n$, every node is assigned a unique subcarrier. Otherwise, a subcarrier is shared by more than one node. The nodes that share the same subcarrier will contend for and access it using a CSMA/CA policy that we will describe in Section IV-B. The subcarrier allocation aims to minimize interference and contention among the nodes. Hence, if two nodes u and v are **hidden** to each other, we aim to assign them different subcarriers (i.e., $g(u) \neq g(v)$), if possible. If two nodes that were hidden to each other are assigned different subcarriers, the hidden node problem is removed. We also should ensure that there is not excessive contention (among the nodes that are in communication range of each other) on some subcarrier. Let $H(u)$ denote the estimated set of nodes that are hidden terminal to u (when using the same subcarrier). Note that the BS is assumed to know the node locations (see Section II). Hence, it can estimate $H(u)$ for any node u based on the locations and estimated communication range of the nodes. Let $\Phi(f_i)$ be the set of nodes that are assigned subcarrier f_i. In the beginning, $\Phi(f_i) = \emptyset$, $\forall i$. For every node u whose subcarrier is not yet assigned, we do the following. We assign it a subcarrier such that $|\Phi(g(u)) \cap H(u)|$ is minimum. If there is more than one such subcarrier, then we assign the one with minimum $|\Phi(g(u))|$. After this assignment, hidden terminals of the associated nodes are updated. Thus, our approach reduces the impact of hidden terminal problem.

B. Transmission Policy

In SNOW, the nodes transmit to the BS using a CSMA/CA approach. It keeps the nodes more flexibility, decentralized, and energy efficient. We do not need to adopt time synchronization, time slot allocation, or to preschedule the nodes. The nodes will sleep (by turning off the radios), and will wake up if they have data to send. After sending the data, a node will go back to sleep again. This will provide high energy-efficiency to the power constrained nodes. We adopt a CSMA/CA policy similar to the one implemented in TinyOS [31] for low power sensor nodes which is very simple (with no RTS/CTS). It uses a static interval for random back-off. Specifically, when a node has data to send, it wakes up by turning its radio on. Then it performs a random back-off in a fixed *initial back-off window*. When the back-off timer expires, it runs CCA (Clear Channel Assessment) and 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. The node then can go to sleep again.

The BS station always remains awake to listen to nodes' requests. The nodes can send whenever they want. There can also be messages from the BS such as management message (e.g., network management, subcarrier reallocation, control message etc.). Hence, we adopt a periodic beacon approach for downward messages. Specifically, the BS periodically sends a beacon containing the needed information for each node through a single message. The nodes are informed of this period. Any node that wants/needs to listen to the BS message can wake up or remain awake (until the next message) accordingly to listen to the BS. The nodes can wake up and sleep autonomously. Note that the BS can encode different data on different subcarriers, carrying different information on different subcarriers if needed, and send all those as a single OFDM message. As explained in Section III-C, the message upon reception will be decoded at the nodes, each node decoding only the data carried in its subcarrier.

C. Handling ACK

Sending ACK after every transmission is crucial but poses a number of **challenges**. **First**, since the nodes asynchronously transmit, if the BS sends ACK after every reception, it may lose many packets from other nodes when it switches to Tx mode. **Second**, the BS uses a wide channel while the node needing ACK uses only a narrow subcarrier of the channel. The AP needs to switch to that particular subcarrier which is expensive as such switching is needed after every packet reception. Note that the BS can receive many packets in parallel and asynchronously. Thus when and how these packets can be acknowledged is a **difficult question**. We adopt a dual radio design at the BS of SNOW which is a practical choice as the BS is power-rich. Thus the BS will have two radios - one for only transmission, called ***Tx radio***, and the other for only reception, called ***Rx radio***. The Tx radio will make all transmissions whenever needed and can sleep when there is no Tx needed. The Rx radio will always remain in receive mode to receive packets. As shown in Figure 5, both radios use the same spectrum and have the same subcarriers - the subcarriers in the Rx radio are for receiving while the same in the Tx radio are for transmitting. Such a dual radio BS design will allow us to enable n concurrent transmissions and receptions. Since each node (non BS) has just a single half-duplex radio, it can be either receiving or transmitting, but not doing both at a time. Thus if k out of n subcarriers are transmitting, the remaining $n-k$ subcarriers can be receiving, thereby making at most n concurrent transmissions/receptions.

Handling ACK and two-way communication using a dual-radio BS still poses the following **challenges**. **First,** while the two radios at the BS are connected in the same module and the Tx radio can send an ACK immediately after a packet is received on the Rx radio, it has to send ACK only to the nodes from which it received packet. Thus some subcarriers will need to have ACK frame while the remaining ones may carry nothing or some data packet. While our PHY design allows to handle this, the challenge is that some ACK/s can

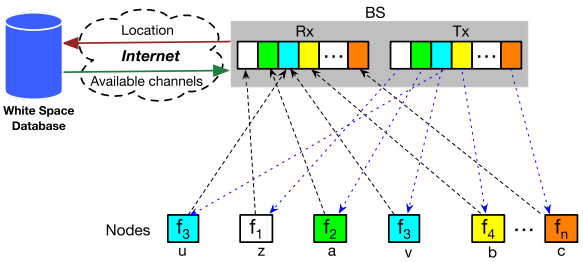


Fig. 5. SNOW architecture with dual radio BS and subcarriers.

be due while the radio is already transmitting some ACK/s. The key question is: *"How can we enable ACK immediately after a packet is received at the BS?"* **Second,** another serious challenge is that the receptions at the Rx radio can be severely interfered by the ongoing transmissions at the Tx radio as both radios operate on the same spectrum and are close to each other. **Third,** ACK on a subcarrier can be interfered if a node sharing it starts transmitting before the said ACK is complete.

D-OFDM allows us to encode any data on any subcarrier while the radio is transmitting. Thus the design allows us to encode any time on any number of subcarriers and enable ACKs to asynchronous transmissions. If there is nothing to transmit, the Tx radio sleeps. Since a node has a single half-duplex radio, it will either transmit or receive. Let us first consider for a subcarrier which is assigned to only one node such as subcarrier f_1 in Figure 5 assigned only to z. Node z will be in receive mode (waiting for ACK) when the Tx radio at the BS sends ACK on f_1. Now consider for a subcarrier which is assigned to more than one node such as subcarrier f_3 in Figure 5 which is assigned to u and v. When u is receiving ACK from the BS, if v needs to transmit it will sense the subcarrier busy and make random back-off. Thus any node sharing a subcarrier f_i will not interfere an ACK on f_i. Hence, transmitting ACK on a subcarrier f_i from the Tx radio has nothing to interfere at f_i of the Rx radio at the BS. Subcarrier f_i at Rx will be receiving the ACK on it sent by the Tx radio and can be ignored by the decoder at the Rx radio. Thus the subcarriers which are encoded with ACKs at the Tx radio will have energy. The remaining ones that are not encoded with ACK/data have no energy. During this time, the nodes may be transmitting on those subcarriers. Thus when the Tx radio transmits, its un-encoded subcarriers cannot interfere the same subcarriers at the Rx radio. The subcarriers carrying ACKs are orthogonal to them and will not interfere either.

D. Other Features of the MAC Protocol

1) Further Mitigating Hidden Terminal Problem: We partially handle hidden terminal problem in subcarrier allocation and MAC protocol. Consider nodes u and v in Figure 5 both of which are assigned subcarrier f_3. Now consider u and v are hidden to each other. When the TX radio of the BS sends ACK to node u that has just made a transmission to the BS, this ACK signal will have high energy on the subcarrier f_3 at the Rx radio of the BS. At this time, if node v makes a transmission to the BS, it will be interfered. Since v will run

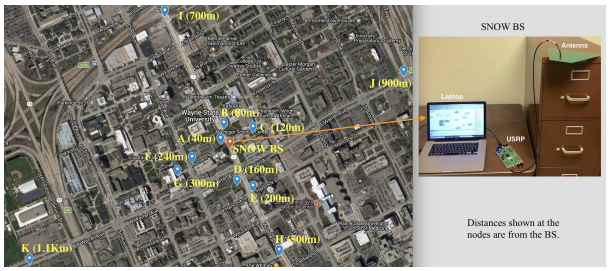


Fig. 6. Node positions in the Detroit metropolitan area.

TABLE I
DEFAULT PARAMETER SETTINGS

Parameter	Value
Frequency Band	572-578MHz
Orthogonal Frequencies	574.4, 574.6, 574.8, 575.0, 575.2, 575.4, 575.6, 575.8MHz
Subcarrier modulation	BPSK
Packet Size	40 bytes
BS Bandwidth	6MHz
Node Bandwidth	400kHz
Spreading Factor	8
Transmit (Tx) Power	0dBm
Receive Sensitivity	-94dBm
SNR	6dB
Distance	1.1km

CCA and sense the energy on f_3 it will not transmit. This result is somewhat similar to that of the CTS frame used in WiFi networks to combat hidden terminal problem.

2) Peer-to-Peer Communication: Two nodes that want to communicate can be hidden to each other or may have different subcarriers. Hence, we adopt peer-to-peer communication through the BS. For example, in Figure 5, if node a wants to send a packet to node b, it cannot send directly as they use different subcarriers. First, a transmits to the BS on subcarrier f_2, and then the BS transmits to b on subcarrier f_4 (in its next beacon when b will wake up if it is sleeping).

3) Handling Various Dynamics: **First**, we handle spectrum dynamics as follows. When the BS's spectrum availability changes due to primary user activity, the BS performs a new spectrum allocation. The nodes whose subcarriers may no more be available may have no way to know the new subcarrier allocation. We handle this by allocating one or more backup subcarriers (similar to backup whitespace channels adopted in [10]). If a node does not receive any beacon for a certain interval, it will assume that its subcarrier is no more available and will switch to a backup subcarrier and wait for BS message. The BS will keep sending rescue information on that backup subcarrier which will be received by that node. For robustness, we maintain multiple backup subcarriers.

Second, we share the loads among the subcarriers by reallocating or swapping. That is, if a subcarrier becomes congested we can un-assign some node from it and assign it a less congested one. **Third**, we allocate some subcarrier for **node joining and leaving**. When a new node wants to join the network, it communicates with the BS on this subcarrier. It can transmit its identity and location to the BS. The BS then assigns it an available subcarrier. Similarly, any node from which the BS has not received any packet for a certain time window can be excluded from the network.

V. IMPLEMENTATION

We have implemented SNOW in GNU Radio [20] using USRP devices [21]. GNU Radio is software-defined radio toolkit [20]. USRP is a hardware platform to transmit and receive for software-defined radio [21]. We have used 9 USRP devices (2 as BS and 7 as SNOW nodes) in our experiment. Two of our devices were USRP B210 while the remaining are USRP B200, each operating on band 70 MHz - 6GHz. The packets are generated in IEEE 802.15.4 structure with random payloads. We implement the decoder at the BS using 64-point G-FFT which is sufficient due to our limited number of devices. In downward communication, multiple parallel packet lines are BPSK modulated on the fly and fed into a *streams-to-vector* block that is fed into IFFT that generates a composite time domain signal.

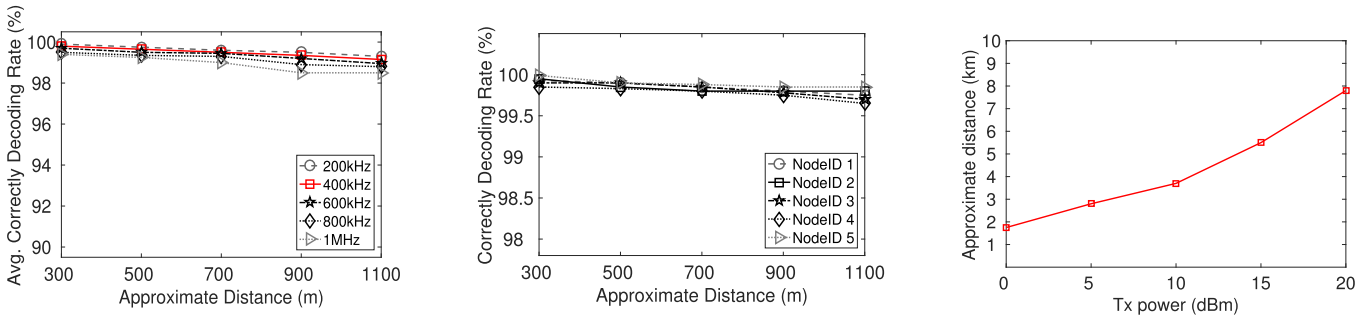
VI. EXPERIMENTS

We deployed SNOW in the Detroit metropolitan area (Michigan), in an indoor environment, and in a rural area of Rolla (Missouri) to observe its performance in various radio environments. In the following subsections, we describe our experimental results in these three deployments. We also compare its performance with existing technologies.

A. Deployment in a Metropolitan City Area

1) Setup: Figure 6 shows different nodes and the BS positions of our deployment in the Detroit metropolitan area. Due to varying distances (max. $\approx$ 1.1km) and obstacles between the BS and these nodes, the SNR of received signals varies across these node positions. We keep all of the antenna heights at 5ft above the ground. Unless mentioned otherwise, Table I shows the default parameter settings for all experiments.

2) Reliability Over Distances and Tx Power: To demonstrate the reliability at various distances, we place the nodes at 300m, 500m, 700m, 900m, and 1100m away from the BS, respectively. At each distance, each node transmits 10,000 packets asynchronously to the BS and vice versa. CDR (which indicates the correctly decoding rate as defined in Section III-E.1) is used as a key metric in our evaluation. Figure 7(a) demonstrates uplink reliability under varying subcarrier bandwidths when the nodes are at different distances from the BS and all transmit at 0dBm. As all nodes transmit at the same Tx power from different distances, the uplink communication in this scenario is subject to *near-far effect*. Namely, the signals at the BS from the nearer transmitters are stronger than those from the farther transmitters, thereby causing packet loss from the formers. This happens because the side-lobes of the stronger signals from nearby nodes may overwhelm the weaker signals from the faraway nodes. In our setup, the maximum difference between the distances of any pair of transmitters from the BS is $\approx$ 1km. Yet, we have observed at least 98% CDR from all transmitters (Figure 7(a)) which indicates that this distance difference is not enough to cause near-far effect. This is reasonable because near-far effect is relatively lesser in D-OFDM, compared to CDMA (Code Division Multiple Access) where it is quite high, due to orthogonality of the signals. It needs more extensive experiments and perhaps very large differences between the



(a) (b) (c)

Fig. 7. Reliability over distances and varying Tx power. (a) Uplink reliability. (a) Uplink reliability. (c) Distance with varying Tx powers.

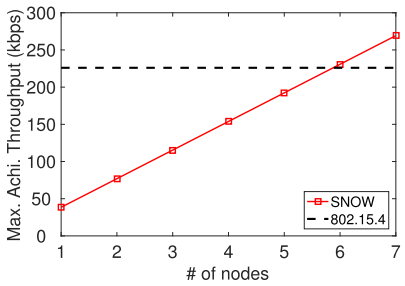


Fig. 8. Maximum achievable throughput.

node distances to observe the effect of near-far problem, which we have not explored in this paper.

Figure 7(b) demonstrates high reliability in downlink under varying distances. As shown at five different nodes for a subcarrier bandwidth of 400kHz, all the nodes can decode more than 99.5% of the packets even though they are 1.1km away from the BS. To demonstrate the feasibility of adopting SNOW in LPWAN, we moved one node much farther away from the BS and vary the Tx power from 0 dBm up to 20 dBm. As shown in Figure 7(c), with 20dBm of Tx power, SNOW BS can decode from approximately 8km away, hence showing its competences as an LPWAN technology.

3) Maximum Achievable Throughput: In this experiment, we evaluate the *maximum achievable throughput* (i.e., maximum total bits that the BS can receive per second) in SNOW. Each node transmits 10,000 packets, each of 40 bytes. In SNOW, after each transmission a node waits for its ACK (hence it does not continuously transmit). Figure 8 shows that SNOW can achieve approximately 270kbps when 7 nodes transmit. We consider the maximum achievable throughput in a typical IEEE 802.15.4 based WSN of 250kbps bit rate as a baseline. Its maximum achievable throughput is shown considering ACK after each transmission. As expected, the number of nodes does not impact its maximum achievable throughput as its BS can receive at most one packet at a time. A channel in the IEEE 802.15.4 based network is much wider than a SNOW subcarrier and has a higher bit rate (250kbps vs 50kbps). Hence, SNOW surpasses the baseline when it has at least 6 nodes. But the SNOW throughput keeps increasing linearly with the number of nodes while that in the baseline remains unchanged. Thus, although we have results for up to 7 nodes, the linear increase in SNOW throughput gives a clear message

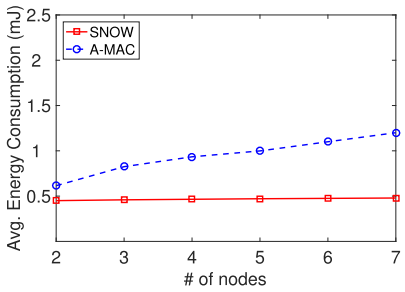
TABLE II
CURRENT CONSUMPTION IN CC1070

Device mode	Current Consumption
Tx	17.5 mA
Rx	18.8 mA
Idle	0.5 mA
Sleep	0.2 μA

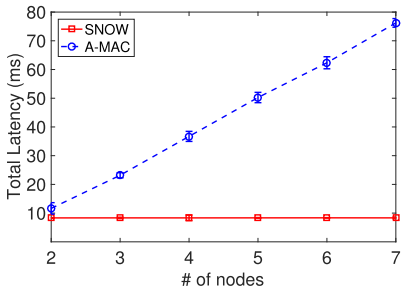
that it is superior in throughput and scalability to any protocol used for traditional WSN.

4) Energy Consumption and Latency: To demonstrate the efficiency in terms of energy and latency, we compare SNOW with a traditional WSN design. Specifically, we consider A-MAC [33] which is an energy efficient MAC protocol for IEEE 802.15.4 based WSN. To estimate the energy consumption and network latency in SNOW nodes, we place 7 nodes each 280m apart from the BS. For a fair comparison, for A-MAC, we place the nodes 40m apart from each other making a linear multi-hop network due to their shorter communication ranges. In both of the networks, we start a convergecast after every 60 seconds. That is, each node except the BS generates a packet every 60 seconds that is ready to be transmitted. Our objective is to collect all the packets at the BS.

Since the USRP devices do not provide any energy consumption information, we use the energy model of CC1070 by Texas Instruments [34]. This off-the-shelf radio chip operates in low frequencies near TV white spaces and also uses BPSK modulation. Table II shows the energy model of CC1070. Since the BS is line-powered, we keep it out of the energy calculation. We run multiple rounds of convergecast for 2 hours in both of the networks. Figure 9(a) shows the average energy consumption in each node per convergecast. Regardless of the number of nodes, on average a SNOW node consumes nearly 0.46mJ energy. On the other hand in A-MAC, on average each node consumes nearly 1.2mJ when 7 nodes participate in convergecast. For a large number of nodes, this value will be very high. Figure 9(b) shows the convergecast latency in both SNOW and A-MAC. We calculate the total time to collect all the packets at the BS from all the nodes counting from the time the packets were generated at the nodes. SNOW takes approximately 8.3ms while A-MAC takes nearly 77ms to collect packets from all 7 nodes. Theoretically, SNOW should take almost constant amount of time to collect all the packets as long as the number of nodes is no greater than that of available subcarriers. Again,

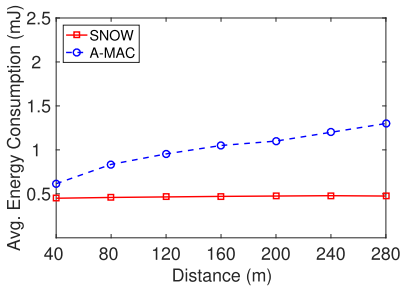


(a)

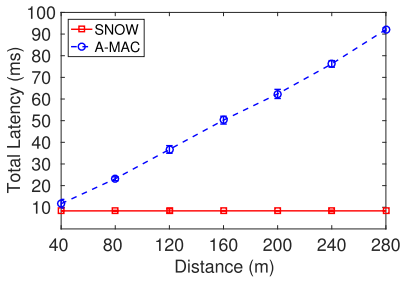


(b)

Fig. 9. Energy consumption and latency in convergecast. (a) Energy consumption. (b) Total latency.



(a)



(b)

Fig. 10. Energy consumption and latency over distance. (a) Energy consumption. (b) Total latency.

due to a small network size, the differences between SNOW and A-MAC are not significant in this experiment.

Energy and latency over distances: Using the same setups as the above, Figure 10 compares energy and latency between SNOW and A-MAC over distances. Figure 10(a) shows that, a node in SNOW consumes on average 0.475mJ of energy to deliver a packet to the BS that is 280m away. On the other hand, an A-MAC node consumes nearly 1.3mJ of energy to deliver one packet to a sink that is 280m away. Also,

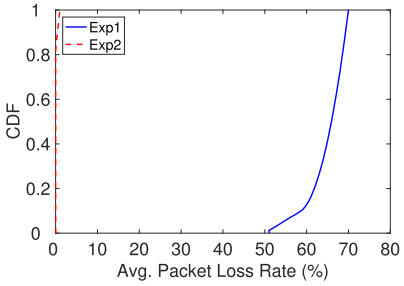


Fig. 11. Performance under hidden terminals.

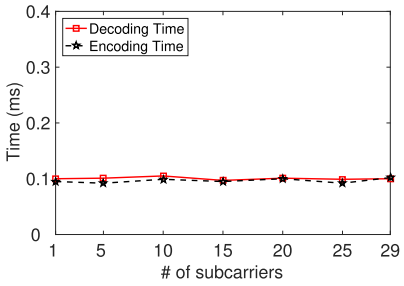
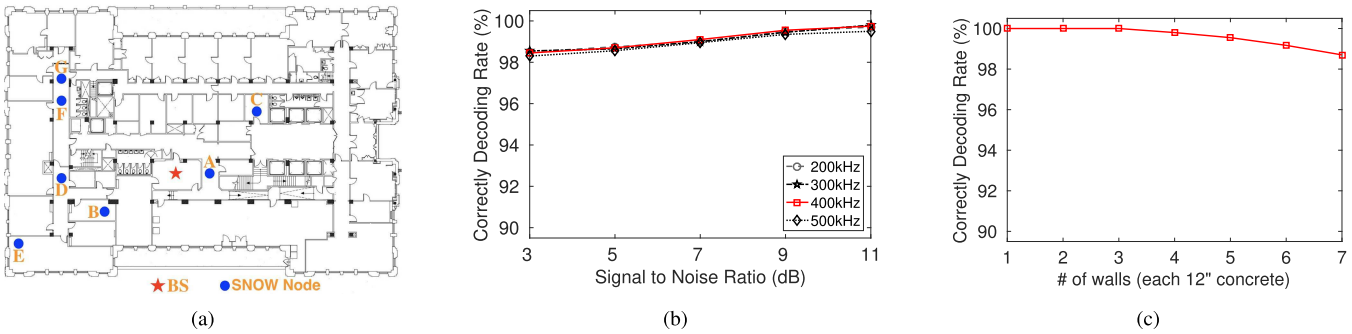


Fig. 12. Encoding and decoding time

Figure 10(b) shows that a SNOW and A-MAC node takes 8.33ms and 92.1ms of latency to deliver one packet to the BS, respectively. As the distance increases, the differences become higher, demonstrating SNOW's superiority.

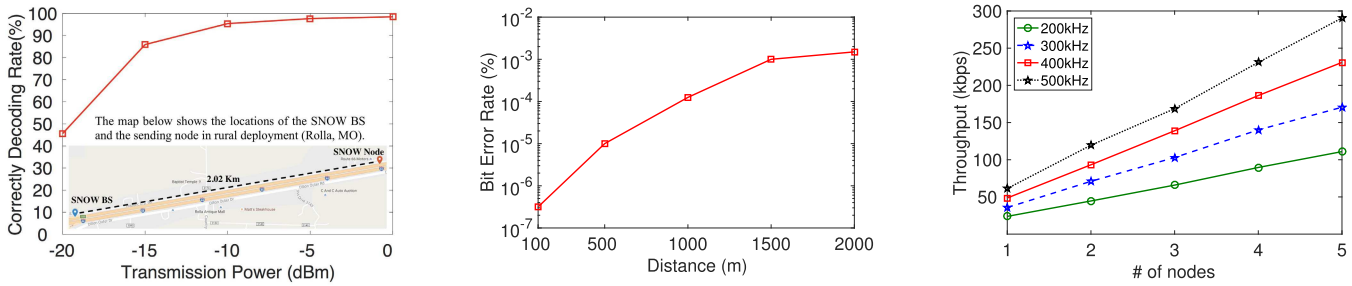
5) Handling Hidden Terminal Problem: To test the performance of SNOW under hidden terminal, we adjust the Tx powers of the nodes at the positions shown in Figure 6 so that (i) nodes A, B and C are hidden to nodes D and E; (ii) D and E are not hidden to each other; (iii) A, B and C are not hidden to each other. We conduct two experiments. In experiment 1 (Exp1), the hidden nodes are assigned the same subcarriers. For example, the BS assigns one subcarrier to node A and D (hidden to each other), another subcarrier to nodes B, D and E (B is hidden to D and E). In experiment 2 (Exp2), the BS assigns different subcarriers to the nodes hidden to each other. Exp2 reflects the SNOW MAC protocol. Each node sends 100 packets to the BS in both experimental setups. After getting the ACK for each packet (or, waiting until ACK reception time), each node sleeps for a random time interval between 0-50ms. After sending 100 packets, each node calculates its packet loss rate and averages it. We repeat this experiment for 2 hours. Figure 11 shows the CDF of average packet loss rate. In Exp1, average packet loss rate is 65% while in SNOW MAC protocol (Exp2) it is 0.9%, which demonstrates the benefits of incorporating location-aware subcarrier allocation.

6) BS Encoding Time and Decoding Time: Although we have seven devices to act as SNOW nodes, we can calculate the data encoding time or decoding time in all 29 subcarriers at the BS as it depends on the number of bins in the IFFT algorithm. Theoretically, the encoding/decoding time for any number of nodes at the BS should be constant as the IFFT/G-FFT algorithm runs with the same number of bins every time. However, we do separate experiments by encoding/decoding data to/from 1 to 29 nodes. We run each experiment for 10 minutes and record the time needed in the worst



(a) (b) (c)

Fig. 13. Reliability in indoor environments. (a) Indoor node positions. (b) Reliability at various SNR. (c) Propagation through walls.



(a) (b) (c)

Fig. 14. Performance of SNOW in rural deployment. (a) Reliability vs Tx power. (b) BER over distances. (c) Throughput vs bandwidth.

case. Figure 12 shows that both encoding time and decoding time are within 0.1ms. This encoding/decoding time is very short as IFFT/G-FFT runs very fast, and is similar to standard encoding/decoding time in WSNs for one packet.

B. Indoor Deployment

1) Setup: Figure 13(a) shows the positions of the SNOW nodes and BS (on floor plan) all on the same floor (293,000 sq ft) of the Computer Science Building at Wayne State University. We fixed the position of the BS (receiver) while changing the positions of the node. In this experiment a node transmits 10,000 consecutive packets at each position.

2) Results: Figure 13(b) shows the CDR over various SNR conditions under varying subcarrier bandwidths. At SNR of 3dB the CDR is around 98.5% for all bandwidths. We observe that increasing the SNR, the CDR increases accordingly for all bandwidths. This is due to the effect of noise, obstacles, and multipath over SNR. Figure 13(c) shows CDR under varying number of walls between sender and receiver. We achieve at least 98.5% CDR when the line of sight is obstructed by up to 7 walls (each $12''$ concrete). SNOW achieves reliable communication even in indoor environments due to low frequency and narrow bandwidth.

C. Deployment in a Rural Area

1) Setup: A rural deployment of SNOW is characterized by two key advantages - higher availability of TV white spaces and longer communication range due to lesser absence of obstacles such as buildings. We deployed SNOW in a rural area of Rolla, Missouri. In this deployment, we used five USRP devices that acted as SNOW nodes. We follow the similar antenna and default parameter setup as described in Section VI-A.1 and Table I.

2) Distance, Reliability, and Throughput: The map embedded in Figure 14(a) shows the locations of the BS and a node 2km away from the BS. The node transmits 1000 40-byte packets consecutively. The same figure shows the reliability (in terms of CDR) of the link under varying Tx power. Specifically, SNOW achieves 2km+ communication range at only 0 dBm Tx power which is almost double that we observed in our urban deployment. This happens due to a cleaner light of sight in the former. Similarly, Figure 14(b) shows the BER at the SNOW BS while decoding packets from various distances. The results show the decodability of the packets transmitted (at 0dBm) from 2km away as BER remains $\leq 10^{-3}$. Like our urban deployment, here also SNOW's maximum achievable throughput linearly increases with the number of nodes (Figure 14(c)).

VII. Comparison With Existing LPWANs

A. SNOW vs LoRa/SIGFOX

LPWANs are emerging as a key technology driving the IoT, with multiple competing technologies being offered or under development. SIGFOX [7] and LoRa [8] are two very recent LPWAN technologies that operate in the unlicensed ISM band. Their devices require to adopt duty cycled transmission of only 1% or 0.1% making them less suitable for many WSNs that involve real-time applications or that need frequent sampling. SIGFOX supports a data rate of 10 to 1,000bps. A message is of 12 bytes, and a device can send at most 140 messages per day. Each message transmission typically takes 3s [35] while SNOW can transmit such a 12-byte message in less than 2ms.

Semtech LoRa modulation employs Orthogonal Variable Spreading Factor (OVSF) which enables multiple spread signals to be transmitted at the same time and on the same channel. OVSF is an implementation of traditional CDMA

where, before transmission, each signal is spread over a wide spectrum range through a user's code. Using 125kHz bandwidth and a LoRa spreading factor (LoRa-SF) of 10, a 10-byte payload packet in LoRa has an air time of 264.2ms typically [36], which is at least 100 times that in SNOW for the same-size message (according to our experiments). The higher the LoRa-SF, the slower the transmission and the lower the bit rate in LoRa. This problem is exacerbated by the fact that large LoRa-SFs are used more often than the smaller ones [37]. According to the LoRa specification [8], its range in urban area is 2–5 km and in rural area is 15–20 km. As Figure 7(c) shows, SNOW range is approximately 8km near urban areas (suburban), showing a similar communication ranges. Some recent studies have however shown that, without line of sight, LoRa communication range is quite small [38], specially indoors where it was found to be at most 100m [39].

1) Scalability Analysis: One important limitation of OVSF is that the users' codes have to be mutually orthogonal to each other, limiting the scalability. LoRa uses 6 orthogonal LoRa-SFs (12 to 7), thus allowing up to 6 different transmissions on a LoRa channel of any bandwidth simultaneously. Using one TV channel (6MHz wide), we can get 29 OFDM subcarriers (each 400kHz) for SNOW which enables 29 simultaneous transmissions. Using a narrower bandwidth like SIGFOX/LoRa would yield even a higher number of subcarriers per channel in SNOW. Using m' white space channels, its number of simultaneous transmissions multiplies by m'.

Scalability of SIGFOX/LoRa is achieved assuming extremely low traffic. For example, if a device sends one packet per hour, a LoRaWAN SX1301 gateway (that uses 8 separate radios) can handle about 62,500 devices [8]. With its 12-byte message and 140 messages per device per day, one SIGFOX gateway can support 1 million devices [7]. We now estimate the scalability of SNOW for this communication scenario. Using one TV channel (6MHz width), we can get 29 OFDM subcarriers (each 400kHz). The total time for a 12-byte message transaction between a SNOW node and the BS is less than 2ms (including Tx-Rx turnaround time). A group of 29 nodes can transmit simultaneously, each on a distinct subcarrier. Note that SNOW uses an asynchronous MAC protocol for flexibility and scalability. We can reduce the MAC protocol to a simple polling scheme to roughly estimate the number of nodes that can can be supported comfortably in a SNOW of a single BS. Specifically, every time we can schedule 29 nodes (n nodes) to transmit simultaneously. If every device sends 140 messages per day (like SIGFOX), every subcarrier can be shared by $\frac{24*3600*1000}{140*2} > 308,571$ devices. Thus 29 subcarriers can be shared by $308,571 * 29 > 8.9$ million devices. If we consider a downward message after every group of simultaneous transmissions by 29 nodes to schedule the next group of transmissions, SNOW with one white space channel can support at least $8.9/2 \approx 4.45$ million devices. Using m' channels, it can support $4.45 * m'$ million devices. This back-of-envelop calculation indicates that SNOW can support a significantly larger number of devices than SIGFOX/LoRa. Next, we will compare their performance in simulations.

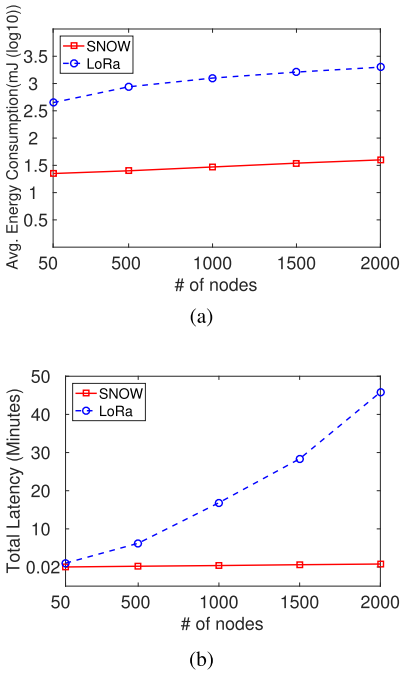


Fig. 15. SNOW vs LoRa. (a) Energy Consumption. (b) Latency.

2) Energy and Latency in Simulation: For large-scale evaluation of SNOW, we perform simulations in QualNet [40]. Since there exists no publicly available specification for SIGFOX, we compare SNOW with LoRa to demonstrate higher efficiency and scalability of SNOW. The simulation setups and results are explained as follows.

a) Setup: We consider a LoRa gateway with 8 parallel demodulation paths, each of 500kHz wide (e.g. Semtech SX1301 [41]). For fair comparison, we choose a BS bandwidth of 500kHz $*$ 8 $=$ 4MHz from white spaces in SNOW and split into 19 overlapping (50%) orthogonal subcarriers, each of 400kHz wide. For each, we create a single-hop star network. All the nodes are within 2km radius of the BS/gateway. We generate various number of nodes in both of the networks. The nodes are distributed evenly in each demodulator path of LoRa gateway. In each demodulator path, LoRa uses the ALOHA protocol. In each network, we perform convergecast. Every node sends 100 40-byte packets with same spreading factor of 8 to the BS/gateway and sleeps for 100ms afterwards. For LoRa, we calculate the airtime of a 40-byte packet (34.94ms) using Lora-calculator [42] and use it in simulation. For its energy profiling, we consider the LoRa iM880B-L [43] radio chip with its minimum supported Tx power of 5dBm.

b) Results: Here we compare SNOW and LoRa in terms of energy consumption and network latency. As Figure 15(a) shows (in $\log_{10}$ scale), for a network of 2000 nodes, the packets are collected at the SNOW BS in 0.79 minutes consuming an average energy of 22.22mJ per node while the LoRa gateway collects those in 45.81 minutes consuming an average energy of 450.56mJ per node. Both energy consumption and latency in SNOW grow extremely slowly. The results indicate their linear (with number of nodes) growth with an extremely small slope as n nodes can transmit in parallel, where n (=19 in this simulation) is the number of subcarriers. Both energy consumption and latency in SNOW are thus much less

compared to LoRa. The MAC protocols in both networks also play role. Our results show that, using the same bandwidth, SNOW can support a much larger number of nodes than LoRa. Thus we have observed the superiority of SNOW over LoRa in terms of scalability, energy consumption, and latency.

B. SNOW vs Other LPWAN Technologies

While OFDM has been adopted for multi-access in the forms of OFDMA and SC-FDMA in various broadband (e.g., WiMAX [44]) and cellular (e.g., LTE [24]) technologies, they rely on strong time synchronization which is very costly for low-power nodes. We adopted OFDM for the first time in WSN design and without requiring time synchronization. D-OFDM enables multiple packet receptions that are transmitted asynchronously from different nodes which was possible as WSN needs low data rate and short packets. To combat fading and to support high data rates, for uplink communication in both OFDMA and SC-FDMA adopted in WiMAX and LTE, respectively, the BS depends on multiple antennas to receive from multiple nodes. Downlink transmissions in both OFDMA and SC-FDMA are made using single antenna. In contrast, D-OFDM enables multiple receptions using a single antenna and also enables different data transmissions to different nodes using a single antenna. Both WiMAX and LTE use OFDMA in downlink direction. WiMAX uses OFDMA in uplink direction also. Due to high peak-to-average power ratio (PAPR), OFDMA in uplink direction may cause high power dissipation of transmitter amplifiers of the low-power nodes, causing lower battery life. SNOW nodes use a single subcarrier and does not suffer from PAPR problem. While SC-FDMA has relatively lower PAPR, to meet the high data rate requirement in LTE (86 Mbps in uplink) and to allow concurrent transmitters its receiver is designed by using multiple antennas at the cost of high energy consumption [24]. Such issues are less severe for low data rate and small packet sizes and we realize D-OFDM with much simpler design.

5G [45] is envisioned to meet IoT use cases using the cellular infrastructure. Currently, the 5G standard is still under development. NB-IoT [46] is a narrowband LPWAN technology standard to operate on cellular infrastructure and bands. Its specification was frozen at Release 13 of the 3GPP specification (LTE-Advanced Pro [47]) in June 2016. These technologies would require devices to periodically wake up to synchronize with the network, giving a burden on battery life. Also, the receiver design to enable multiple packet receptions simultaneously using SC-FDMA requires **multiple antennas**. Note that setting up multiple antennas is **difficult** for lower frequencies as the antenna form factor becomes large. The antennas need to be spaced $\lambda/2$ apart, where λ is the wavelength. Doing this is difficult as λ is **large** for lower frequencies, and even more difficult and expensive to do this for every sector to be served by the base station. Having low data rate and small packets, SNOW PHY design remains much simpler and both the transmitters and the receiver can have a single antenna and the BS can receive multiple packets simultaneously using single antenna radio. Furthermore, the SNOW MAC has several novel features including a location-aware spectrum allocation for mitigating hidden terminal problems, per-transmission ACK for asynchronous transmissions, and the capability of handling peer-to-peer communication. Another important advantage of SNOW is that it exploits white spaces which have widely available free spectrum, while the above LPWANs operate in the licensed band or limited ISM band.

VIII. Other Related Work

Prior work focused on accessing white spaces through spectrum sensing [48] or geo-location approach [49] for broadband service. A review of those work can be found in [11]. In contrast, the objective of the SNOW design is to exploit white spaces for designing highly scalable, low-power, wide-area WSN. The upcoming IEEE 802.15.4m [50] standard aims to exploit white spaces as an extension to IEEE 802.15.4. SNOW can therefore help shape and evolve such standards.

IX. Conclusions

We have proposed the design of SNOW. It is the first low power, long range WSN over TV white spaces to support reliable, asynchronous, bi-directional, and concurrent communication between numerous sensors and a base station.

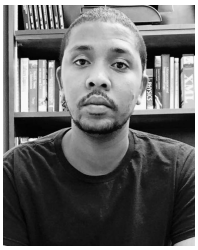
References

[1] R. N. Murty *et al.*, "CitySense: An urban-scale wireless sensor network and testbed," in *Proc. HST*, 2008, pp. 583–588.
[2] (2017). *Oilfield Automation and Monitoring*. [Online]. Available: http://www.petrocloud.com/solutions/oilfield-monitoring/
[3] D. Vasisht *et al.*, "Farmbeats: An IoT platform for data-driven agriculture," in *Proc. NSDI*, 2017, pp. 515–529.
[4] (2003). *IEEE 802.15.4*. [Online]. Available: http://standards.ieee.org/about/get/802/802.15.html
[5] (1997). *IEEE 802.11*. [Online]. Available: http://www.ieee802.org/11
[6] (1998). *Bluetooth*. [Online]. Available: http://www.bluetooth.com
[7] (2009). *SigFox*. [Online]. Available: http://sigfox.com
[8] (2012). *LoRa*. [Online]. Available: https://www.lora-alliance.org
[9] *FCC Second Order*, document ET Docket No FCC 10-174, Sep. 2010.
[10] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with Wi-Fi like connectivity," in *Proc. SIGCOMM*, 2009, pp. 27–38.
[11] X. Ying *et al.*, "Exploring indoor white spaces in metropolises," in *Proc. MobiCom*, 2013, pp. 255–266.
[12] (2017). *Microsoft 4AFRIKA*. [Online]. Available: http://www.microsoft.com/africa/4afrika/
[13] (2013). *Africa Forum*. [Online]. Available: https://sites.google.com/site/tvwsafrica2013/
[14] (2014). *IEEE 802.11af*. [Online]. Available: http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11af-white-fi-tv-space.php
[15] (2011). *IEEE 802.22*. [Online]. Available: http://www.ieee802.org/22/
[16] (2017). *IEEE 802.19*. [Online]. Available: http://www.ieee802.org/19/
[17] A. Saifullah *et al.*, "SNOW: Sensor network over white spaces," in *Proc. SenSys*, 2016, pp. 272–285.
[18] A. Saifullah *et al.*, "Enabling reliable, asynchronous, and bidirectional communication in sensor networks over white spaces," in *Proc. SenSys*, 2017, pp. 9-1–9-14.
[19] Q. Li *et al.*, "MIMO techniques in WiMAX and LTE: A feature overview," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 86–92, May 2010.
[20] (2001). *GNU Radio*. [Online]. Available: http://gnuradio.org
[21] (2015). *USRP B210*. [Online]. Available: http://www.ettus.com/product/details/UB210-KIT
[22] A. Saifullah *et al.*, "CapNet: A real-time wireless management network for data center power capping," in *Proc. RTSS*, 2014, pp. 334–345.
[23] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007.
[24] (2014). *The LTE Standard*. [Online]. Available: https://www.qualcomm.com/media/documents/files/the-lte-standard.pdf
[25] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "A case for adapting channel width in wireless networks," in *Proc. SIGCOMM*, 2008, pp. 135–146.

[26] C.-T. Chen, *System and Signal Analysis*. Stamford, CT, USA: Thomson, 1988.
[27] (2015). *Understanding FFTs and Windowing*. [Online]. Available: http://www.ni.com/white-paper/4844/en/
[28] E. Sourour, H. El-Ghoroury, and D. McNeill, "Frequency offset estimation and correction in the IEEE 802.11a WLAN," in *Proc. VTC*, 2014, pp. 4923–4927.
[29] J.-J. van de Beek *et al.*, "A time and frequency synchronization scheme for multiuser OFDM," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 11, pp. 1900–1914, Nov. 1999.
[30] D. Ismail, M. Rahman, A. Saifullah, and S. Madria, "RnR: Reverse & replace decoding for collision recovery in wireless sensor networks," in *Proc. SECON*, 2017, pp. 1–9.
[31] (2012). *TinyOS*. [Online]. Available: http://www.tinyos.net
[32] (2009). *IEEE 802.15.4c*. [Online]. Available: https://ieeexplore.ieee.org/document/4839293/
[33] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. SenSys*, 2010, pp. 1–4.
[34] (2005). *CC1070*. [Online]. Available: http://www.ti.com/product/CC1070
[35] (2017). *What is SigFox?* [Online]. Available: http://www.link-labs.com/what-is-sigfox/
[36] (2013). *LoRa Modem Design Guide*. [Online]. Available: http://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf
[37] F. Adelantado *et al.*, "Understanding the limits of loRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Jan. 2017.
[38] M. Cattani, C. A. Boano, and K. Römer, "An experimental evaluation of the reliability of LoRa long-range low-power wireless communication," *J. Sens. Actuator Netw.*, vol. 6, no. 2, p. 7, 2017.
[39] N. Vatcharatiansakul, P. Tuwanut, and C. Pornavalai, "Experimental performance evaluation of LoRaWAN: A case study in Bangkok," in *Proc. JCSSE*, 2017, pp. 1–4.
[40] (2013). *QualNet*. [Online]. Available: http://web.scalable-networks.com/content/qualnet
[41] (2015). *Semtech*. [Online]. Available: http://www.semtech.com/wireless-rf/rf-transceivers/sx1301/
[42] SemTech. (2015). *LoRa Calculator by Semtech*. [Online]. Available: http://www.sx1272-lora-calculator.software.informer.com/
[43] (2012). *LoRa iM880B-L*. [Online]. Available: http://www.wireless-solutions.de/products/radiomodules/im880b-l
[44] R. Prasad and F. J. Velez, "OFDMA WiMAX physical layer," in *WiMAX Networks*. Dordrecht, The Netherlands: Springer, 2010, pp. 63–135.
[45] (2017). *Next Generation Mobile Network*. [Online]. Available: http://www.ngmn.org
[46] (2017). *NBIoT*. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete
[47] (2017). *LTE Advanced Pro*. [Online]. Available: https://www.qualcomm.com/invention/technologies/lte/advanced-pro
[48] D. Liu, Z. Wu, F. Wu, Y. Zhang, and G. Chen, "FIWEX: Compressive sensing based cost-efficient indoor white space exploration," in *Proc. MobiHoc*, 2015, pp. 1–10.
[49] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "SenseLess: A database-driven white spaces network," in *Proc. DySpan*, 2011, pp. 10–21.
[50] (2017). *IEEE 802.15.4m*. [Online]. Available: http://www.ieee802.org/15/pub/TG4m.html

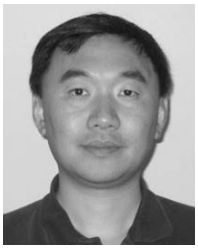
Abusayeed Saifullah received the Ph.D. degree from the Computer Science and Engineering Department, Washington University in St. Louis, in 2014. He is currently an Assistant Professor of computer science with Wayne State University. His research interests concern cyber-physical systems, Internet of Things, and real-time systems. He received the Turner Dissertation Award of Washington University, the Best Student Paper Award at RTSS 2011, the Best Paper Nomination at the IEEE RTAS 2012, the Best Paper Award at the IEEE RTSS 2014, and the Best Paper Nomination at ACM SenSys 2016.

Mahbubur Rahman received the bachelor's degree in computer science and engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Wayne State University. He has co-primary-authored a paper that was nominated for the best paper at ACM SenSys 2016. His research interests include low-power wide-area networks, Internet of Things, and real-time systems.



Dali Ismail received the B.Sc. degree from the University of Science and Technology, Sudan, and the M.Sc. degree in computer science and engineering from Washington University in St. Louis. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Wayne State University. His primary research focuses on low-power wide-area networks used in the emerging Internet of Things.

Chenyang Lu (F'16) received the Ph.D. degree from the University of Virginia in 2001. He is currently the Fullgraf Endowed Chair Professor with Washington University in St. Louis. He has authored or coauthored over 170 research papers with over 18 000 citations and an h-index of 60, and he is one of the most productive authors at top conferences in the embedded and real-time systems area. His research interests include Internet of Things, real-time and embedded systems, and cyber-physical systems. His current work focuses on real-time cloud, industrial cyber-physical systems, and Internet of Medical Things. He is currently the Chair of the IEEE Technical Committee on Real-Time Systems. He served as the Editor-in-Chief for the *ACM Transactions on Sensor Networks* from 2011 to 2017.



Jie Liu (F'18) received the bachelor's and master's degrees from the Department of Automation, Tsinghua University, Beijing, China, and the Ph.D. degree from the Department of Electrical Engineering and Computer Sciences, UC Berkeley, in 2001. From 2001 to 2004, he was a Research Scientist with the Palo Alto Research Center. He is currently the Partner Research Manager and the GM of the Ambient Intelligent Team, Microsoft AI Perception and Mixed Reality. He holds over 100 patents. His research interests root in sensing and interacting with the physical world through computing. He is an ACM Distinguished Scientist and a Distinguished Speaker. He has received six best paper awards in top conferences. He was the chair of a number of top-tier conferences and their steering committees. He was an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING. He is an Associate Editor of the *ACM Transactions on Sensor Networks*.

Ranveer Chandra received the Ph.D. degree from Cornell University. He is currently a Principal Researcher with Microsoft Research, where he is involved in incubation on IoT applications. He is also leading the FarmBeats Project, battery research, and TV white space projects with Microsoft Research. He has published over 80 papers, and filed over 100 patents, with over 80 granted by USPTO. He has received several awards, including the Best Paper Awards at ACM CoNext 2008, World Technology Network in 2012, ACM SIGCOMM 2009, IEEE RTSS 2014, USENIX ATC 2015, and Runtime Verification 2016, the Microsoft Research Graduate Fellowship, the Microsoft Gold Star Award, the MIT Technology Review's Top Innovators Under 35, TR35 in 2010, and a fellow in Communications.