

Enabling Reliable, Asynchronous, and Bidirectional Communication in Sensor Networks over White Spaces

Abusayeed Saifullah*

Wayne State University, Detroit, MI
saifullah@wayne.edu

Mahbubur Rahman*

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

Dali Ismail

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

Chenyang Lu

Washington University, St Louis, MO
lu@cse.wustl.edu

Jie Liu

Microsoft Research, Redmond, WA
jie.liu@microsoft.com

Ranveer Chandra

Microsoft Research, Redmond, WA
ranveer@microsoft.com

ABSTRACT

Low-Power Wide-Area Network (LPWAN) heralds a promising class of technology to overcome the range limits and scalability challenges in traditional wireless sensor networks. Recently proposed Sensor Network over White Spaces (SNOW) technology is particularly attractive due to the availability and advantages of TV spectrum in long-range communication. This paper proposes a new design of SNOW that is asynchronous, reliable, and robust. It represents the first highly scalable LPWAN over TV white spaces to support reliable, asynchronous, bi-directional, and concurrent communication between numerous sensors and a base station. This is achieved through a set of novel techniques. This new design of SNOW has an OFDM based physical layer that adopts robust modulation scheme and 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 MAC protocol that (1) efficiently implements per-transmission acknowledgments of the asynchronous transmissions by exploiting the adopted OFDM design; (2) combines CSMA/CA and location-aware spectrum allocation for mitigating hidden terminal effects, thus enhancing the flexibility of the nodes in transmitting asynchronously. Hardware experiments through deployments in three radio environments - in a large metropolitan city, in a rural area, and in an indoor environment - as well as large-scale simulations demonstrated that the new SNOW design drastically outperforms other LPWAN technologies in terms of scalability, energy, and latency.

CCS CONCEPTS

• **Networks** → **Network protocols**; • **Computer systems organization** → **Sensor networks**;

*Co-first author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SenSys '17, November 6–8, 2017, Delft, Netherlands

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-5459-2/17/11...\$15.00

<https://doi.org/10.1145/3131672.3131676>

KEYWORDS

White space, LPWAN, Sensor Network, OFDM, MAC Protocol.

ACM Reference Format:

Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Jie Liu, and Ranveer Chandra. 2017. Enabling Reliable, Asynchronous, and Bidirectional Communication in Sensor Networks over White Spaces. In *Proceedings of SenSys '17, Delft, Netherlands, November 6–8, 2017*, 14 pages. <https://doi.org/10.1145/3131672.3131676>

1 INTRODUCTION

Sensor networking over TV white spaces has gained interest recently [52, 59, 63]. Wireless sensor network (WSN) in large-scale and wide-area applications (e.g., urban sensing [41], civil infrastructure monitoring [25], oil field management [47], precision agriculture [26]) often needs to connect thousands of sensors over long distances. Due to their short communication range, the existing WSN technologies in the ISM band such as IEEE 802.15.4 [17], 802.11 [15], and Bluetooth [5] cover a large area with numerous devices as multi-hop mesh networks at the expense of energy, cost, and complexity. These limitations can be overcome by letting WSNs operate over TV white spaces. Such a network architecture is called *Sensor Network Over White Spaces (SNOW)*.

White spaces refer to the allocated but locally unused TV spectra, and can be used by unlicensed devices [45, 46]. 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 [46] to learn about unoccupied TV channels at a location. Similar regulations are adopted in many countries. Compared to IEEE 802.15.4 or Wi-Fi, they offer a large number of and less crowded channels, each 6MHz, available in both rural and urban areas [3, 9, 22, 38, 44, 60, 68, 69]. Thanks to their lower frequencies (54 – 862MHz in the US), white spaces have excellent propagation characteristics over long distance and obstacles. Long range will reduce many WSNs to a star-topology that has potential to avoid the complexity, overhead, and latency associated with many-hop mesh networks. Such a paradigm shift must also deal with the challenges that stem from the 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.

Exploiting white spaces for sensor networking is the goal of the on-going IEEE 802.15.4m standardization effort [59]. As an early research effort in this space, we proposed the first design of

SNOW in [52], referred to as **SNOW 1.0** in this paper, to address some of the above challenges. It was designed based on D-OFDM, a distributed implementation of Orthogonal Frequency Division Multiplexing (OFDM), that allowed its base station (BS) to receive multiple packets in parallel. The BS uses wide white space spectrum which is split into narrowband orthogonal subcarriers. Each sensor node is assigned a subcarrier on which it transmits. Despite its promise, SNOW 1.0 has several important limitations as follows.

- (1) D-OFDM in SNOW 1.0 is not implemented for *bi-directional* communication over different subcarriers. Its BS can receive packets from multiple nodes in parallel but *cannot* concurrently transmit different packets to different nodes.
- (2) SNOW 1.0 cannot support per-transmission acknowledgment (ACK) which limits its *reliability*.
- (3) It does not support fully asynchronous operation as the nodes can transmit asynchronously only if their number is no greater than that of the subcarriers. It schedules transmissions from multiple sensors sharing the same subcarriers based on Time-Division Multiple Access (TDMA), which limits their flexibility in transmitting asynchronously.
- (4) It uses amplitude-shift-keying (ASK) which provides simplicity but is not a robust modulation scheme.

In this paper, we address the above challenges and important limitations of SNOW 1.0, and propose a new design of SNOW, referred to as **SNOW 2.0**, that is asynchronous, reliable, and robust. Throughout this paper, with ‘SNOW’ we shall mean SNOW 2.0. The terms ‘SNOW 2.0’ and ‘SNOW 1.0’ will be used when we need to distinguish between this new design and the earlier one. SNOW 2.0 is the first design of a highly scalable, low power, and long range WSN over TV white spaces which is fully asynchronous and enables reliable massive parallel and asynchronous receptions with a single antenna-radio and multiple concurrent data transmissions with a single antenna-radio. This is achieved through a full-fledged physical layer (PHY) design by implementing D-OFDM for multiple access in both directions and through a reliable, light-weight Media Access Control (MAC) protocol. While OFDM has been embraced for multiple access in various wireless broadband and cellular technologies recently (see Section 2.2), its adoption in low power, low data rate, narrowband, and WSN design remains quite new. Taking the advantage of low data rate and short packets, we adopt OFDM in WSN through a much simpler and energy-efficient design. The BS’s wide white space spectrum is split into narrowband orthogonal subcarriers that D-OFDM uses to enable parallel data streams to/from the distributed nodes from/to the BS. SNOW 2.0 thus represents a promising platform for many cyber-physical systems and Internet of Things (IoT) applications that depend on bidirectional sensor data (e.g., Microsoft’s FarmBeats in IoT for agriculture [63]).

The specific contributions of this paper are as follows.

- We design a D-OFDM based PHY for SNOW with the following features for enhanced scalability, low power, long range. (1) It adopts robust modulation scheme such as Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK). (2) Using a single antenna-radio, the BS can receive concurrent transmissions made by the sensor nodes asynchronously. (3) Using a single antenna-radio, the BS can send different data to different nodes concurrently.

Note that the above design is different from MIMO radio adopted in various wireless domains such as LTE, WiMAX, 802.11n [27] as the latter uses multiple antennas to enable multiple transmissions and receptions.

- We develop a lightweight MAC protocol for operating the nodes with greater freedom, low power, and reliability. The SNOW MAC 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 and concurrent 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 key features include the capability of handling peer-to-peer communication, spectrum dynamics, load balancing, and network dynamics.
- We implement SNOW in GNU Radio [13] using Universal Software Radio Peripheral (USRP) [49] 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 just one packet.
- We perform experiments through SNOW deployments in three different radio environments - a large metropolitan city, a rural area, an indoor testbed - as well as simulations. All results demonstrate the superiority of SNOW over several LPWAN technologies in terms of scalability, latency, and energy. Large-scale simulations show a 100% increase in SNOW throughput while having both latency and energy consumption half compared to our earlier design.

In the rest of the paper, Section 2 overviews related work. Section 3 describes the model. Section 4 presents the PHY. Section 5 presents the MAC protocol. Sections 6, 7, and 8 present implementation, experiments, and simulations, resp. Section 9 is the conclusion.

2 RELATED WORK

2.1 White Spaces Network

To date, the potential of white spaces is mostly being tapped into for broadband access by industry leaders such as Microsoft [1, 50] and Google [12]. Various standards bodies such as IEEE 802.11af [16], IEEE 802.22 [20], and IEEE 802.19 [19] are modifying existing standards to exploit white spaces for broadband access. In parallel, the research community has been investigating techniques to access white spaces through spectrum sensing [4, 23, 24, 29] or geo-location approach [11, 14, 35, 40, 70] mostly for broadband service. A review of white space networking for broadband access can be found in [52, 68]. In contrast, the objective of our work is to exploit white spaces for designing highly scalable, low-power, long range, reliable, and robust SNOW. We proposed SNOW 1.0 in [52]. As already pointed out in Section 1, SNOW 1.0 does not support bidirectional, reliable, and fully asynchronous communication. Hence, it is not a suitable platform for applications that need ACK, sensing and control [51, 53], or bidirectional sensor data [63]. Our proposed new SNOW design overcomes all of these limitations and achieves enhanced scalability, reliability, and robustness.

2.2 Low-Power Wide-Area Network (LPWAN)

2.2.1 SNOW vs LoRa/SIGFOX. LPWAN technologies are gaining momentum in recent years, with multiple competing technologies being offered or under development. SIGFOX [57] and LoRa [6, 32, 37, 64] are two very recent LPWAN technologies that operate in 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 3 seconds [28] while SNOW can transmit such a 12-byte message in less than 2ms as we experimented in [52].

Semtech LoRa modulation employs Orthogonal Variable Spreading Factor (OVSF) which enables multiple spread signals to be transmitted at the same time on the same channel. OVFSF is an implementation of traditional Code Division Multiple Access (CDMA) where before each signal is transmitted, the signal is spread over a wide spectrum range through the use of a user's code. Using 125kHz bandwidth and LoRa spreading factor (LoRa-SF) of 10, a 10-byte payload packet in LoRa has an air time of 264.2ms typically [31], which is at least 100 times that in SNOW for the same size message [52]. 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. For instance, as studied in [2], considering a scenario with end-devices distributed uniformly within a round-shaped area centered at the gateway, and a path loss calculated with the Okumura-Hata model [39] for urban cells, the probability that an end-device uses a LoRa-SF of 12 would be 0.28, while that of 8 would be 0.08.

One important limitation of OVFSF is that the users' codes have to be mutually orthogonal to each other, limiting the scalability of the network that adopts this technique. LoRa uses 6 orthogonal LoRa-SFs (12 to 7), thus allowing up to 6 different transmissions on a channel simultaneously. Using one TV channel (6MHz wide), we can get 29 OFDM subcarriers (each 400kHz) for SNOW which enables 29 simultaneous transmissions on a single TV channel. Using a narrower bandwidth like SIGFOX/LoRa would yield even a higher number of subcarriers per channel in SNOW. Note that white spaces can consist of more than one TV channel. Using M channels, the number of simultaneous transmissions multiplies by M in SNOW. Hence, our back-of-envelope calculation even for SNOW 1.0 in [52] showed its superiority in scalability over SIGFOX/LoRa. Since there exists no publicly available specification for SIGFOX, we compare SNOW with LoRa in Section 8 through simulation to demonstrate higher efficiency and scalability of SNOW.

2.2.2 Comparison with The Other LPWAN Technologies.

SNOW achieves high scalability by exploiting the existing OFDM technology for multi-access. OFDM is a well-known modulation technique and it has been adopted for multi-access in various forms in various wireless broadband and cellular technologies recently. However, its usage in low-power, low-rate, narrowband and wireless sensor network domain is still new. Our adopted technique, D-OFDM, in SNOW has similarity with several OFDM multiple access techniques such as OFDMA (Orthogonal Frequency Division Multiple Access) and SC-FDMA (Single Carrier Frequency Division

Multiple Access) adopted in WiMAX [48, 65] and LTE [34, 67, 71]. For uplink communication in both OFDMA and SC-FDMA adopted in WiMAX and LTE, respectively, the BS uses **multiple antennas to receive from multiple nodes**. 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. OFDMA is known to be more sensitive to a null in the channel spectrum and it requires channel coding or power/rate control to overcome this deficiency. Specifically, for its usage in uplink communication, the transmit power of the senders need to be adjusted so that the received signal strengths from different senders are close. In low power network, this becomes difficult. Also, OFDMA has a high peak-to-average power ratio (PAPR) which leads to difficulties in transceiver design [34, 48, 67, 71]. This also implies high power consumption and lower battery life for the sending nodes in uplink communication. Therefore, the 3GPP standardization group has decided to use SC-FDMA instead in LTE for uplink communication [34, 67, 71].

While SC-FDMA has relatively lower PAPR, to meet the high data rate requirement in LTE (86 Mbps in uplink), its receiver design for allowing multiple simultaneous transmitters is complicated, and is designed by using multiple antennas at the cost of high energy consumption [34, 67, 71]. Such issues are less severe for low data rate and small packet sizes and we can realize with much simpler design. Therefore, our similar design, D-OFDM, remains much simpler and multiple receptions and multi-carrier transmission can be done using a single antenna of the radio in SNOW.

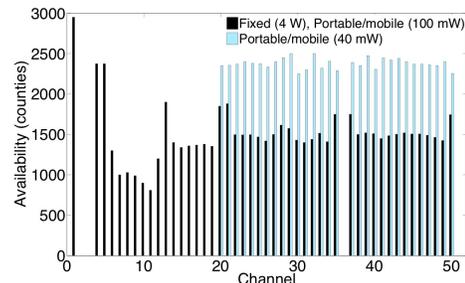


Figure 1: White space in the US counties [58]: showing the number of counties (y-axis) where the channels (x-axis) are white space.

5G [43] is envisioned to meet IoT use cases in addition to telecommunications applications using the cellular infrastructure. Currently, the 5G standard is still under development. NB-IoT [42] 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 [33]) 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 due to lower frequency. 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 packet sizes, 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. We also design a complete MAC protocol for SNOW which features 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, spectrum dynamics, load balancing, and network dynamics. Another important advantage of SNOW is that it is designed to exploit white spaces which have widely available free spectrum (as shown in Figure 1), while the above standards are designed to use licensed band or limited ISM band.

3 SYSTEM MODEL

WSNs are characterized by small packets, low data rate, and low power [52]. The nodes are typically battery powered. Thus, scalability and energy-efficiency are the key concerns in WSN design. We consider a WSN where a lot of sensor nodes are associated with a BS. Each sensor node (called ‘**node**’ throughout the paper) is equipped with a single half-duplex narrow-band radio operating in the white space spectrum. Due to long communication range even at low power (e.g., several kilometers at 0 dBm transmission power in our experiment in Section 7) 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 2. 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.

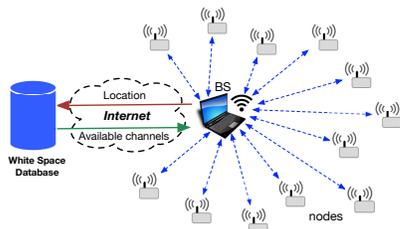


Figure 2: The network structure of SNOW.

The BS uses a wide channel split into subcarriers, each of equal spectrum width (bandwidth). Each node is assigned one subcarrier on which it transmits to and receives from the BS. 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 other nodes very simple and energy-efficient. The nodes do not do spectrum sensing or cloud access. The BS determines white spaces by accessing a cloud-hosted database through the Internet as shown in Figure 2. We assume that it knows the locations of the nodes either through manual configuration or through some existing WSN localization techniques such as those based on ultrasonic sensors or other sensing modalities [36]. Localization is out of scope of this paper. The BS selects white space channels that are available at its own location and at the locations of all other nodes. We use two radios at the BS to support concurrent transmission and reception as described in Section 5.

4 PHYSICAL 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; (3) support robust modulation such as BPSK.

4.1 Design Rationale

For scalability and energy efficiency, we design the PHY based on D-OFDM. OFDM is a frequency-division multiplexing scheme that uses a large number of closely spaced orthogonal subcarrier signals to carry data on multiple parallel data streams between a sender and a receiver. As discussed before, it has been adopted for multi-user in various forms in various wireless broadband and cellular technologies recently. D-OFDM is a distributed implementation of OFDM introduced in [52] for multi-user access. Unlike OFDMA and SC-FDMA for multi-access, D-OFDM enables multiple receptions using a single antenna and also enables different data transmissions to different nodes using a single antenna.

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 [8]. Thus, 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 BPSK which is highly robust due to difference of 180° between two constellation points, and is widely used (e.g. in WiMAX 16d, 16e; WLAN 11a, 11b, 11g, 11n). Since BPSK and QPSK are fundamentally similar with the latter being less robust with higher bit rate, with minor modification QPSK (which is used in IEEE 802.15.4 at 2.4 GHz [17]) is also adoptable in SNOW.

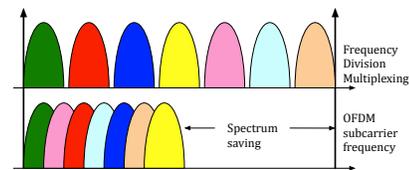


Figure 3: Typical frequency-division multiplexing vs OFDM.

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, they are *orthogonal* to each other. Two sinusoids with frequencies that are integer multiples of a common one satisfy this criterion. The orthogonal subcarriers can be **overlapping**, thus increasing the spectral efficiency (as shown in Figure 3). As long as orthogonality is maintained, it is still possible to recover the individual subcarriers’ signals despite their overlapping spectrums. Specifically, in the **downward communication** in SNOW (i.e. 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** in SNOW (i.e. when many nodes transmit on different subcarriers to a single

Subcarrier 1	Subcarrier 2	...	Subcarrier n
...	$b_{1,2}$
...	$b_{2,2}$...	$b_{1,n}$
...	$b_{3,2}$...	$b_{2,n}$
$b_{1,1}$	$b_{4,2}$...	$b_{3,n}$
$b_{2,1}$	$b_{5,2}$...	$b_{4,n}$
$b_{3,1}$	$b_{6,2}$...	$b_{5,n}$
$b_{4,1}$	$b_{7,2}$...	$b_{6,n}$
$b_{5,1}$	$b_{8,2}$...	$b_{7,n}$
$b_{6,1}$	$b_{9,2}$...	$b_{8,n}$
\vdots	\vdots	\vdots	\vdots

Figure 4: 2D matrix for decoding in upward communication

radio of 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.

Note that if the BS radio has n subcarriers it can receive from at most n nodes simultaneously. Similarly, it can carry at most n different data at a time. When the number of nodes is larger than n , a subcarrier is shared among multiple nodes and their communication is governed by the MAC protocol (Section 5). To explain the PHY design we ignore subcarrier allocation and consider only the n nodes who have occupied the subcarriers for transmission.

4.2 Upward Communication

Here we describe how we enable parallel receptions at a single radio at the BS when each node's data is modulated based on BPSK or QPSK. In our D-OFDM design, 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.

Decoding upon Distributed Encoding:

Every node independently encodes based on BPSK (or QPSK) the data on its subcarrier. To decode a composite OFDM signal generated from orthogonal subcarriers from the distributed nodes, we adopt *Global FFT Algorithm (G-FFT)* which runs FFT on the entire range of the spectrum of the BS, instead of running a separate FFT for each subcarrier. To receive asynchronous transmissions, the BS keeps running the G-FFT algorithm. A vector v of size equal to the number of FFT bins stores the received time domain samples. The G-FFT is performed on v at every cycle of the baseband signal. For n subcarriers, we apply an m point G-FFT algorithm, where $m \geq n$. 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 also use a phase threshold that represents maximum allowable phase deviation in the received samples. For BPSK, one symbol is mapped into one bit, where in QPSK one symbol is mapped to a dibit. Since any node can transmit any time without any synchronization, the correct 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. Figure 4 shows the 2D matrix where entry $b_{i,j}$ represents i -th bit (for BPSK) of j -th subcarrier. The same process thus repeats. We handle spectral leakage through the *Blackman-Harris windowing* [62].

4.3 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. Such a communication goal is challenging due to asymmetric bandwidth between the transmitter (BS in this case) and the receivers (the nodes in this case). In the following, we describe our approach to achieve this in SNOW.

Encoding for Distributed Decoding:

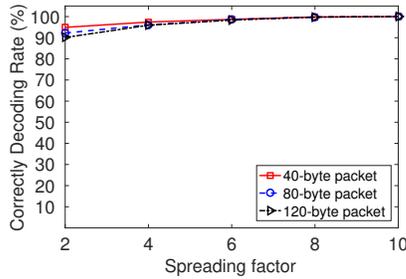
Our design approach based on D-OFDM is to enable distributed demodulation at the nodes without any coordination among them. That is, from the received OFDM signal, every node will independently decode based on BPSK/QPSK the data from the signal component on its subcarrier only. In our approach, the main design technique lies in the encoding part at the BS. We enable this by adopting IFFT (Inverse FFT) at the transmitter side that encodes different data on different subcarriers. 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 the IFFT algorithm. 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.

4.4 Using Fragmented Spectrum

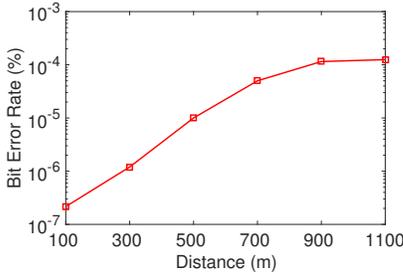
White space spectrum may be found fragmented. When we cannot find consecutive white space channels while needing more, we may use non-consecutive white spaces. The G-FFT and IFFT algorithms will be run on the entire spectrum (as a single wide channel) that includes all fragments (including the occupied TV channels between the fragments). 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.

4.5 Design Considerations

4.5.1 Link parameters. *Bit spreading* is a technique to reduce bit errors by transmitting redundant bits for ease of decoding in noisy environments. It is widely used in many wireless technologies such as IEEE 802.15.4 [17] and IEEE 802.11b [15]. 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



(a) CDR under varying SF and packet sizes

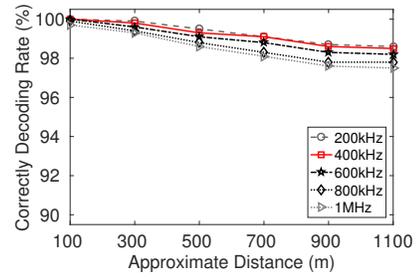


(b) BER over distances when SF=8

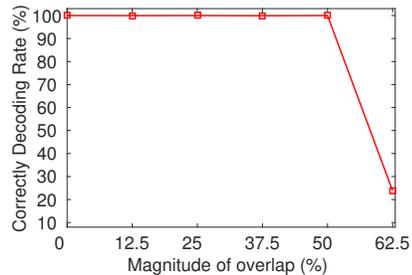
Figure 5: Determining spreading factor

90% of the packets when the sender is 1.1km away and transmits at 0 dBm (Figure 5(a)). Figure 5(b) shows that bit error rate (BER) remains negligible under varying distances (tested up to 1.1km in this experiment). Note that for wireless communications, a packet is usually dropped if its BER exceeds 10^{-3} [21]. Thus we will use 8 as default SF. Since the subcarriers can often violate orthogonality in practice, in our low data rate communication using a spreading factor of 8 helps us mitigate its effects and still recover most of the bits. We have tested the feasibility of different packet sizes (Figure 5(a)). WSN packet sizes are usually short. For example, TinyOS [61] (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.

4.5.2 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 [18]), 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 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 6(a) shows the feasibility of various bandwidths. In our experiments, 400kHz bandwidth provides our required bit rate under



(a) Reliability over distance



(b) Magnitude of overlap between subcarriers

Figure 6: Determining subcarriers

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 (Figure 6(b)). In our low data rate communication using a spreading factor of 8 helps us mitigate the effects of any orthogonality violation.

5 RELIABLE MAC PROTOCOL

We develop a low overhead MAC protocol for operating the nodes with greater freedom, low power, and reliability. As the nodes transmit asynchronously to the BS, implementing ACK for every transmission is extremely 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 in concurrent and asynchronous scenario. 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.

5.1 Location-Aware Spectrum Allocation

This BS spectrum is split into n overlapping orthogonal subcarriers, 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), the total number of orthogonal subcarriers

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

For example, when $\alpha = 50\%$, $W=6$ MHz, $w=400$ kHz, we can have $n = 29$ orthogonal subcarriers. Let us denote the subcarriers by f_1, f_2, \dots, f_n . 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 make some subcarrier available or unavailable. Since our PHY design is capable of handling fragmented spectrum, such dynamism at the MAC layer is feasible.

The subcarrier allocation is done at the BS. Each node is assigned one subcarrier. Let $f(u)$ 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 subcarrier allocation will also try to minimize interference as well as contention among the nodes sharing the same subcarrier. Hence, our first goal is to try to assign different subcarriers to a pair of nodes that are **hidden** to each other. That is, if two nodes u and v are hidden to each other, we try to meet the condition $f(u) \neq f(v)$. Our second goal is to ensure there is not excessive contention (among the nodes that are in communication range of each other) on some subcarrier compared to others. Let $H(u)$ denote the estimated set of nodes that are hidden terminal to u . Note that the BS is assumed to know the node locations either through manual configuration or through some existing WSN localization techniques such as those based on ultrasonic sensors or other sensing modalities [36]. Localization is out of the scope of this paper. The BS can estimate $H(u)$ for any node u based on the locations and estimated communication range of the nodes. Let the set of nodes that have been assigned subcarrier f_i be denoted by $\Omega(f_i)$. In the beginning, $\Omega(f_i) = \emptyset, \forall i$. For every node u whose subcarrier has not been assigned, we do the following. We assign it a subcarrier such that $|\Omega(f(u)) \cap H(u)|$ is minimum. If there is more than one such subcarrier, then we select the one with minimum $|\Omega(f(u))|$. This will reduce the impact of **hidden terminal problem**.

5.2 Transmission Policy

In SNOW, the nodes transmit to the BS using a CSMA/CA approach. This approach gives us more flexibility and keeps the management more decentralized and energy efficient. Specifically, 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 turn the radios on (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 simple CSMA/CA approach without any RTS/CTS frames. We will adopt a CSMA/CA policy similar to the one implemented in TinyOS [61] for low power sensor nodes that 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 4.3, the message upon reception will be decoded in a distributed fashion at the nodes, each node decoding only the data carried in its subcarrier.

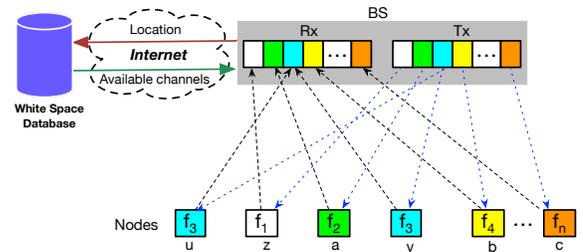


Figure 7: SNOW architecture with dual radio BS & subcarriers

5.3 Reliability

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 7, 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 the 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, handling ACKs for asynchronous transmissions is a difficult problem in wireless domain. The radio needs 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 be due while the radio is already transmitting some ACK/s. That is, while sending some ACK/s another packet's reception can be complete making its ACK due immediately. 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.

Handling the above Challenges in SNOW:

D-OFDM allows us to encode any data on any subcarrier while the radio is transmitting. Thus the design will allow us to encode any time on any number of subcarriers and enable ACKs to asynchronous transmissions. When there is nothing to transmit, the Tx radio can sleep. 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 7 which is assigned only to node 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 7 which is assigned to two nodes, u and v . When u is receiving ACK from the BS, if v attempts to transmit it will sense the subcarrier busy due to BS's ACK on it 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 subset of the subcarriers which are encoded with ACKs at the Tx radio will have energy. The remaining subcarriers that are not encoded with ACK or anything will have no energy due to OFDM design on the signal coming out from the Tx radio of the BS. During this time, the nodes may be transmitting on those subcarriers. Thus when the Tx radio transmits, its un-encoded subcarriers will have no energy and will not be interfering the same subcarriers at the Rx radio. Thus receptions on those subcarriers at the Rx radio can continue without interference. The subcarriers carrying ACKs are orthogonal to them and will not interfere either.

5.4 Other Features of The MAC Protocol

5.4.1 Partially Handling Hidden Terminal. We partially handle hidden terminal problem in subcarrier allocation and MAC protocol. Consider nodes u and v in Figure 7 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 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. Specifically, based on the ACK frame sent by the BS, node v decides not to transmit to avoid interference from the ACK of u 's transmission.

5.4.2 Peer-to-Peer Communication. Two nodes that want to communicate can be hidden to each other or may have different subcarriers. Hence we realize peer-to-peer communication through the BS. For example, in Figure 7, if node a wants to send a packet to b , it cannot send directly as they use different subcarriers. Hence, a will first transmit to the BS on subcarrier f_2 , and then the BS will transmit on subcarrier f_4 to node b in its next beacon time.

5.4.3 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 get the new subcarrier allocation from the BS. We handle this by allocating one or more backup subcarriers (similar to backup whitespace channels adopted in [3]). If a node does not receive any beacon for a certain number of times, it will determine that its subcarrier is no more available and will switch to a backup subcarrier and wait for BS message. The BS will keep sending this rescue information on that backup subcarrier which will thus be received by that node. For robustness, we maintain multiple such backup subcarriers. Another scenario can be the case when some subcarrier becomes overly noisy. To handle this, we adopt **subcarrier swapping** among the nodes. The swapping will be done between bad ones only, not between good ones, not between good and bad ones (as some good subcarrier for a node may become bad after swapping). Exchanging between two nodes who are experiencing a high loss can result in good link quality.

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 adopt **node joining and leaving** by allocating some subcarriers for this purpose. When a new node wants to join the network, it uses this join subcarrier to communicate with the BS. It can transmit its identity and location to the BS. The BS then checks the available white space and 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.

6 IMPLEMENTATION

We have implemented SNOW in GNU Radio [13] using USRP devices [49]. GNU Radio is software-defined radio toolkit [13]. USRP is a hardware platform to transmit and receive for software-defined radio [49]. We have used 9 USRP devices (2 at the 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 modulated on the fly and fed into a *streams-to-vector* block that is fed into IFFT that generates a composite time domain signal.

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

Table 1: Default parameter settings

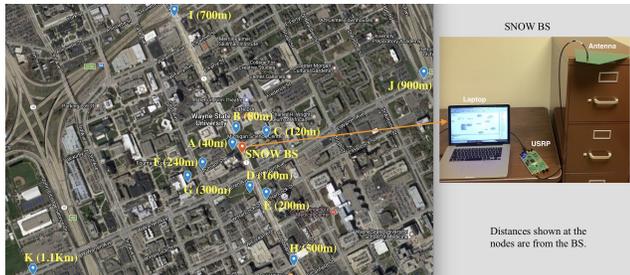


Figure 8: Node positions in the Detroit metropolitan area.

7 EXPERIMENTS

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

7.1 Deployment in A Metropolitan City Area

7.1.1 Setup. Figure 8 shows different nodes and the BS positions in this setting in the Detroit Metropolitan Area. Due to varying distances (max. ≈ 1.1 km) 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 approximately 5ft above the ground. Unless mentioned otherwise, Table 1 shows the default parameter settings for all of the experiments.

7.1.2 Reliability over Distances and Tx Power. To demonstrate the reliability at various distances, we place all 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 4.5.1) is used as a key metric in our evaluation. Figure 9(a) demonstrates uplink reliability under varying subcarrier bandwidths when the nodes are at different distances from the BS and all transmit at 0dBm. Specifically, with 400kHz of subcarrier bandwidth, the BS can decode on average 99.15% of packets from all of the nodes that are 1.1km away. Also, for all other subcarrier bandwidths, the average CDR at the BS stays above 98.5% at all distances. Similarly, Figure 9(b) demonstrates high reliability in downlink under varying distances. As shown at five different nodes for subcarrier bandwidth of 400kHz, all the nodes can decode more than 99.5% of the packets even though they are 1.1km apart from the BS.

With 0dBm (maximum in WSNs based on IEEE 802.15.4) of Tx power and receiver sensitivity of -94dBm (typical sensor devices), we limited our maximum distance between the BS and a node to 1.1km with high reliability. 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 9(c), with 20 dBm of Tx power, SNOW BS can decode from approximately 8km away, hence showing its competences for LPWAN technologies.

7.1.3 Maximum Achievable Throughput. In this experiment, we compare the *maximum achievable throughput* (i.e., maximum total bits that the BS can receive per second) between the new SNOW design (SNOW 2.0) and the earlier design (SNOW 1.0). We add ACK capability to SNOW 1.0 in its downward phase where the BS will switch to each node's subcarrier one after another and send an ACK for all transmissions the BS received in the last upward phase from that node. As soon as all ACKs are sent, SNOW 1.0 will switch to upward phase for receiving again from the nodes as we want to measure its maximum achievable throughput by adding ACK. In SNOW 1.0, the upward phase duration was set to 10s. In both of the networks, each node transmits 10,000 40-byte packets. In SNOW 2.0, after each transmission a node waits for its ACK (hence it does not continuously transmit).

Figure 10 shows that SNOW 2.0 achieves approximately 270kbps compared to 220kbps in SNOW 1.0 when 7 nodes transmit. For better understanding of the maximum achievable throughput, we also draw a baseline, maximum achievable throughput in a typical IEEE 802.15.4 based WSN of 250kbps bit rate. 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. Note that 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, both SNOW 2.0 and SNOW 1.0 surpass the baseline when the number of nodes is 7 or more. 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 that it is superior in throughput and scalability to any protocol used for traditional WSN. Due to a small number of nodes, the throughput improvement of SNOW 2.0 over SNOW 1.0 is not well-visible. Later, in simulation, we show that SNOW 2.0 significantly outperforms SNOW 1.0 in terms of throughput.

7.1.4 Energy Consumption and Latency. To demonstrate the efficiency in terms of energy and latency, we compare SNOW 2.0 with a traditional WSN design. Specifically, we consider A-MAC [10] which is an energy efficient MAC protocol for IEEE 802.15.4 based WSN that operates on 2.4GHz band. To estimate the energy consumption and network latency in SNOW 2.0 nodes, we place 7 nodes each 280m apart from the BS. To compare fairly, we place A-MAC 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 immediately. Our objective is to collect all the packets at the BS.

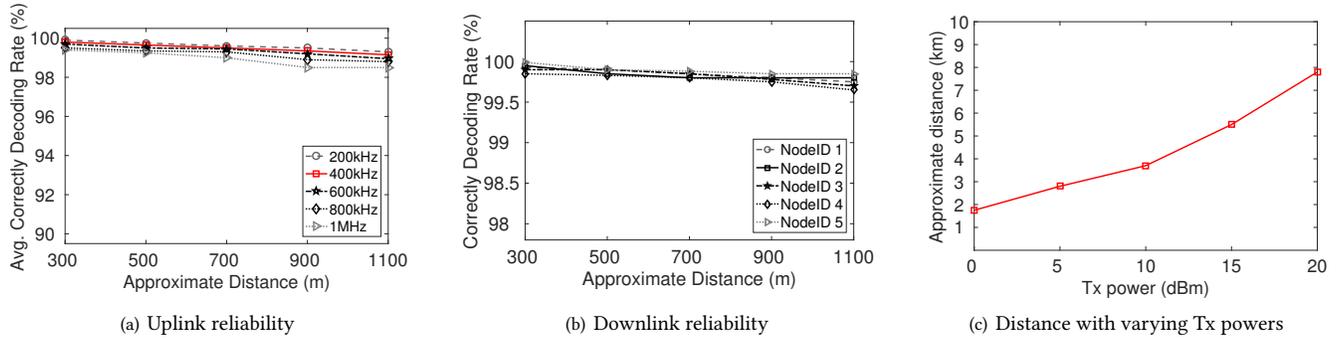


Figure 9: Reliability over distances and varying Tx power.

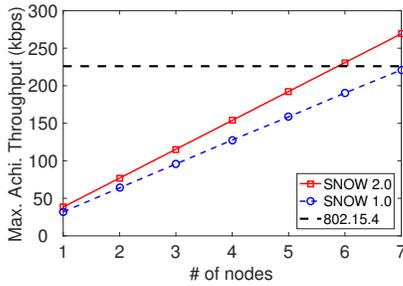
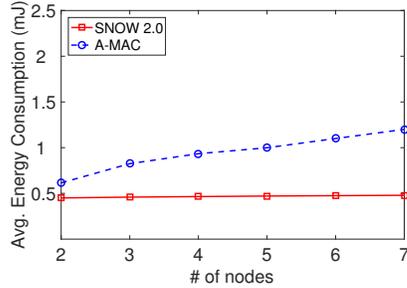
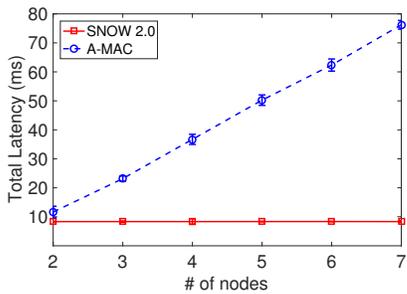


Figure 10: Maximum achievable throughput



(a) Energy consumption



(b) Total latency

Figure 11: Energy consumption and latency in convergecast

Since the USRP devices do not provide any energy consumption information, we use the energy model of CC1070 by Texas Instruments [7]. This off-the-shelf radio chip operates in low frequencies near TV white spaces and also uses BPSK modulation. Table 2 shows

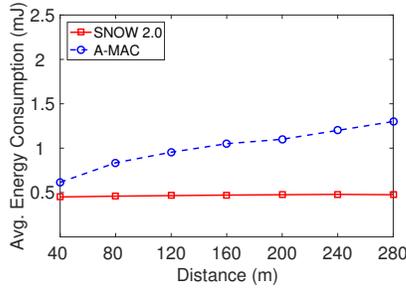
the energy model of CC1070. Since the BS is line-powered, we keep it out of the energy calculation. We run multiple rounds of convergecast for 2 hours in both of the networks. Figure 11(a) shows the average energy consumption in each node per convergecast. Regardless of the number of nodes, on average a SNOW 2.0 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. In practice, with a large number of nodes, A-MAC node consumes significant amount of energy as we found in [52]. Figure 11(b) shows the convergecast latency in both SNOW 2.0 and A-MAC. We calculate the total time to collect all the packets at the BS from all the nodes. SNOW 2.0 takes approximately 8.3ms while A-MAC takes nearly 77ms to collect packets from all 7 nodes. Theoretically, SNOW 2.0 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, due to a small network size, the differences between SNOW 2.0 and A-MAC are not significant in this experiment.

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

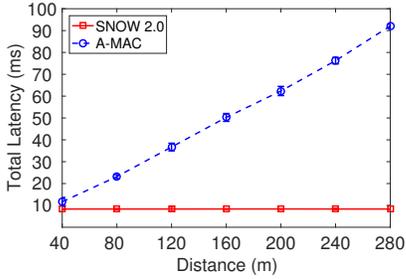
Table 2: Current consumption in CC1070

Energy Consumption and Latency over Distances. With the same setups from previous Section 7.1.4, Figure 12 demonstrates the energy and latency comparison between SNOW 2.0 and A-MAC with respect to distances. Figure 12(a) shows that, a node in SNOW 2.0 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, Figure 12(b) shows that a SNOW 2.0 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.

7.1.5 Handling Hidden Terminal Problem. To test the performance of SNOW 2.0 under hidden terminal, we adjust the Tx powers of the nodes at the positions shown in Figure 8 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



(a) Energy consumption



(b) Total latency

Figure 12: Energy consumption and latency over distance

nodes are assigned the same subcarriers. For example, 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 2.0 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 we average it. We repeat this experiment for 2 hours. Figure 13 shows the CDF of average packet loss in experiments 1 and 2. In Exp1, average packet loss rate is 65%, for SNOW 2.0 MAC protocol (Exp2) it is 0.9%, which demonstrates the benefits of combining location-aware subcarrier allocation in SNOW 2.0 MAC.

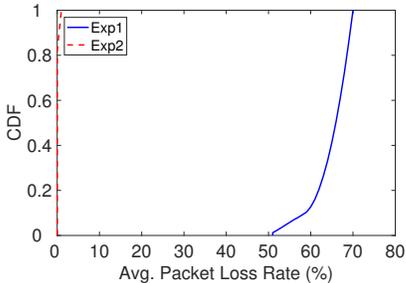


Figure 13: Performance under hidden terminals

7.1.6 BS Encoding Time and Decoding Time. While we have seven USRP devices to act as SNOW nodes we can calculate the data encoding time or decoding time in all 29 subcarriers (in a 6MHz

TV channel) 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 case. Figure 14 shows that both encoding time and decoding time are within 0.1ms. This encoding/decoding time is very fast as IFFT/G-FFT runs very fast. Thus our BS encoding/decoding time is almost similar to standard encoding/decoding time for one packet in typical WSN devices.

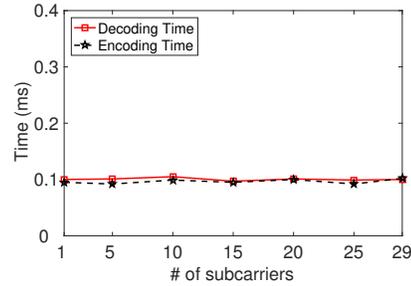


Figure 14: Encoding and decoding time at BS

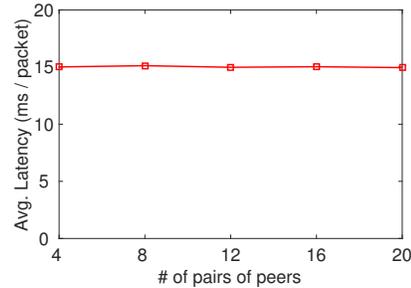


Figure 15: Peer-to-peer avg. packet delivery time

7.1.7 Handling Parallel Peer-to-Peer Communication. In this experiment, we aim to show the feasibility of parallel peer-to-peer communications in SNOW 2.0. This kind of scenarios are common in wireless control [66]. Having seven SNOW nodes, we generate different numbers of pairs of peers. In each pair of peers, one node delivers 1000 40-byte packets to the other via BS. Figure 15 shows that the average latency for one peer-to-peer packet delivery remains within 15ms. While we tested up to 20 pairs, we can expect similar latency as long as the number of pairs \leq the number of subcarriers. Thus, SNOW 2.0 can be a feasible platform even for applications that rely on peer-to-peer communication.

7.2 Indoor Deployment

7.2.1 Setup. Figure 16(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.

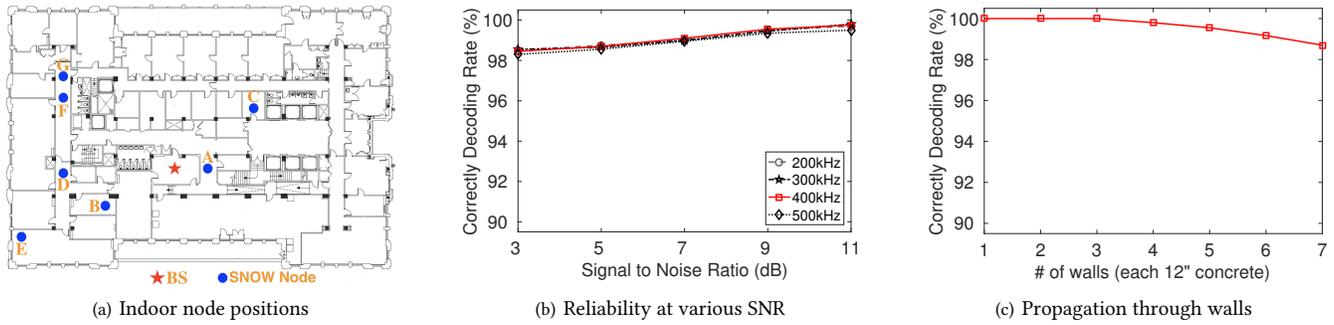


Figure 16: Reliability in indoor environments

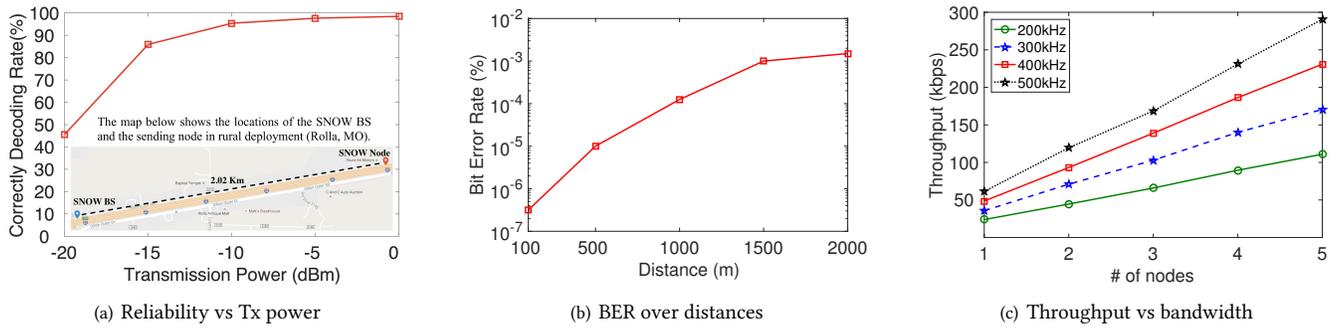


Figure 17: Performance of SNOW 2.0 in rural deployment

7.2.2 **Results.** Figure 16(b) shows the CDR over various SNR conditions under varying subcarrier bandwidths. At SNR of 3dB the CDR is around 98.5% for all subcarrier bandwidths. We observe that while increasing the SNR, the CDR increases accordingly for all subcarrier bandwidth. This is due to the effect of noise, obstacles, and multipath over SNR. Figure 16(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). Due to low frequency and narrow bandwidth, SNOW 2.0 can reliably communicate in indoor environments.

7.3 Deployment in A Rural Area

7.3.1 **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 2.0 in a rural area of Rolla, Missouri. We used five USRP devices that acted as SNOW nodes. We follow the similar antenna and default parameter setup as described in Section 7.1.1 and Table 1.

7.3.2 **Distance, Reliability, and Throughput.** The map embedded in Figure 17(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 2.0 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 17(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 $\leq 10^{-3}$. As expected like in our urban deployment, here also SNOW's maximum achievable throughput linearly increases as we increase the number of nodes (Figure 17(c)).

8 SIMULATION

For large-scale evaluation of SNOW 2.0, we perform simulations in QualNet [54]. We compare its performance with SNOW 1.0 and LoRa [32]. Note that both SNOW 1.0 and SNOW 2.0 take the advantage of wide white space spectrum while LoRa operates in limited ISM band. Hence, for a fair comparison, we compare SNOW 2.0 with SNOW 1.0 and LoRa separately under different setups.

8.1 Comparison with SNOW 1.0

8.1.1 **Setup.** For both SNOW 2.0 and SNOW 1.0, we consider 81MHz of BS bandwidth and split it into 400 overlapping (50%) orthogonal subcarriers each of 400kHz wide. We create a single-hop star network for both. Nodes are distributed within 2km radius of the BS. Then we use a setup similar to Section 7.1.3 for SNOW 2.0 and SNOW 1.0 MAC protocols. Here, the upward phase duration for SNOW 1.0 was set to 1s. In both networks, each node sends 100 40-byte packets and we calculate the throughputs at the BS, average energy consumption per node, and total time needed to collect all packets. As SNOW 1.0 cannot enable per-transmission ACK, we include ACK in SNOW 1.0 after completing upward phase for fair comparison. Thus, when a node sends a packet to its BS the node waits until the end of upward period to receive the ACK.

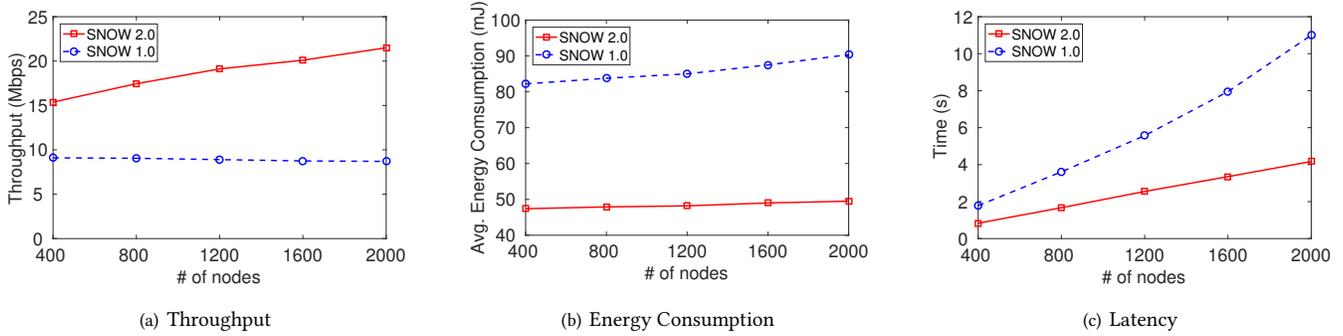


Figure 18: SNOW 2.0 vs SNOW 1.0

8.1.2 Results. Figure 18(a) shows that the throughput of SNOW 2.0 is almost double that of SNOW 1.0 under varying number of nodes. Throughput in SNOW 1.0 increases slowly due to the longer downward communication cycles for delivering all the ACKs. In contrast, SNOW 2.0 can deliver per transmission ACK to each asynchronous transmission and its throughput increases almost linearly with increase in the number of nodes. (We acknowledge that SNOW 1.0 throughput would be similar to ours without ACK but using ACK for wireless communication is quite critical and for a fair comparison we include ACK in SNOW 1.0.) For the same reason, both the energy consumption and latency in SNOW 1.0 are almost two times that in SNOW 2.0 (Figures 18(b) and 18(c)). This demonstrates the superiority of SNOW 2.0 over SNOW 1.0 in terms of scalability.

8.2 Comparison with LoRa

8.2.1 Setup. We consider a LoRa gateway with 8 parallel demodulation paths, each of 500kHz wide (e.g. Semtech SX1301 [56]). For fair comparison, we choose a BS bandwidth of $500kHz * 8 = 4MHz$ from white spaces in SNOW 2.0 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 pure ALOHA MAC 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 [55] and use it in simulation. For its energy profiling, we consider the LoRa iM880B-L [30] radio chip with its minimum supported Tx power of approximately 5dBm.

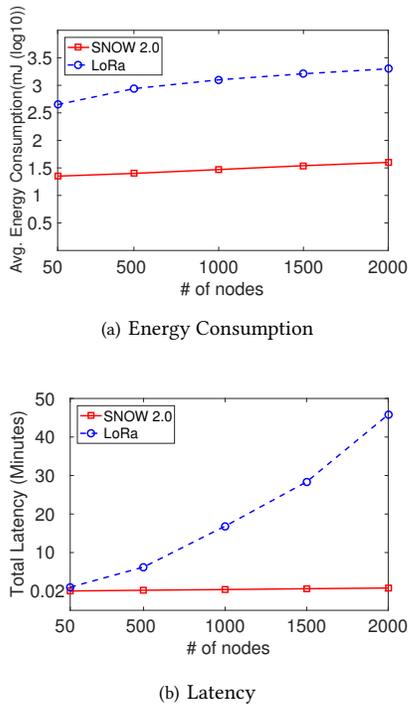


Figure 19: SNOW 2.0 vs LoRa

8.2.2 Results. As the superiority of SNOW 1.0 over LoRa in terms of throughput was numerically demonstrated in [52] and we have already demonstrated the superiority of SNOW 2.0 over SNOW 1.0 in Section 8.1, here we compare them only in terms of energy consumption and latency. As shown in Figure 19(a) (in log₁₀ scale), for a 2000-node network, the packets are collected at the SNOW BS within 0.79 minutes consuming 22.22mJoule of average energy per node while that are collected at the LoRa gateway within 45.81 minutes consuming 450.56mJoules of average energy per node. Both energy consumption and latency in SNOW 2.0 are much less since it allows 19 nodes to transmit in parallel, while only 8 nodes can transmit concurrently in LoRa. The MAC protocols in both networks also play role. Our results show that, using the same bandwidth, SNOW 2.0 can support a larger set of nodes.

9 CONCLUSION

In this paper, we have proposed the design of an asynchronous, reliable, and robust Sensor Network over White spaces (SNOW). This new design of SNOW represents the first low power and long range sensor network over TV white spaces to support reliable, asynchronous, bi-directional, and concurrent communication between numerous sensors and a base station. Hardware experiments through deployments in multiple geographical areas as well as simulations demonstrated that it significantly outperforms the state-of-the-art designs in terms of scalability, energy, and latency.

ACKNOWLEDGMENTS

This work was supported by NSF through grants CRII-1565751 (NeTS), CNS-1320921 (NeTS) and 1646579 (CPS), and by the Fullgraf Foundation.

REFERENCES

- [1] Microsoft 4AFRIKA. 2017. (2017). <http://www.microsoft.com/africa/4afrika/>.
- [2] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the Limits of LoRaWAN. *IEEE Communications Magazine* (January 2017).
- [3] Paramvir Bahl, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. 2009. White space networking with wi-fi like connectivity. *ACM SIGCOMM Computer Communication Review* 39, 4 (2009), 27–38.
- [4] Raamkumar Balamurthy, Harshit Joshi, Cong Nguyen, Ahmed K Sadek, Stephen J Shellhammer, and Cong Shen. 2011. A TV white space spectrum sensing prototype. In *New frontiers in dynamic spectrum access networks (DySPAN), 2011 IEEE symposium on*. IEEE, 297–307.
- [5] Bluetooth [n. d.]. ([n. d.]). <http://www.bluetooth.com>.
- [6] Martin Bor, Utz Roedig, Thiemo Voigt, and Juan Alonso. 2016. Do LoRa low-power wide-area networks scale? (2016).
- [7] CC1070 [n. d.]. ([n. d.]). <http://www.ti.com/product/CC1070>.
- [8] Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra, and Paramvir Bahl. 2008. A Case for Adapting Channel Width in Wireless Networks. In *SIGCOMM '08*.
- [9] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li. 2009. Mining spectrum usage data: a large-scale spectrum measurement study. In *MobiCom '09*.
- [10] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and Andreas Terzis. 2010. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *SenSys '10*.
- [11] X. Feng, J. Zhang, and Q. Zhang. 2011. Database-assisted multi-AP network on TV white spaces: Architecture, spectrum allocation and AP discovery. In *DySPAN '11*.
- [12] TV White Spaces Africa Forum. 2013. (2013). <https://sites.google.com/site/twvsafrica2013/>.
- [13] GNU Radio [n. d.]. ([n. d.]). <http://gnuradio.org>.
- [14] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch. 2008. Geolocation database techniques for incumbent protection in TV. In *DySPAN '08*.
- [15] IEEE 802.11 [n. d.]. ([n. d.]). <http://www.ieee802.org/11>.
- [16] IEEE 802.11af [n. d.]. ([n. d.]). <http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11af-white-fi-tv-space.php>.
- [17] IEEE 802.15.4 [n. d.]. ([n. d.]). <http://standards.ieee.org/about/get/802/802.15.html>.
- [18] IEEE 802.15.4c [n. d.]. ([n. d.]). <https://standards.ieee.org/findstds/standard/802.15.4c-2009.html>.
- [19] IEEE 802.19 [n. d.]. ([n. d.]). <http://www.ieee802.org/19/>.
- [20] IEEE 802.22 [n. d.]. ([n. d.]). <http://www.ieee802.org/22/>.
- [21] Dali Ismail, Mahbubur Rahman, Abusayed Saifullah, and Sanjay Madria. 2017. RnR: Reverse & Replace Decoding for Collision Recovery in Wireless Sensor Networks. In *SECON '17*.
- [22] V.D. Jaap, R. Janne, A. Andreas, and M. Petri. 2011. UHF white space in Europe: a quantitative study into the potential of the 470-790 MHz band. In *DySPAN '11*.
- [23] H. Kim and K. G. Shin. 2008. Fast Discovery of Spectrum Opportunities in Cognitive Radio Networks. In *DySPAN '08*.
- [24] H. Kim and K. G. Shin. 2008. In-band Spectrum Sensing in Cognitive Radio Networks: Energy Detection or Feature Detection?. In *MobiCom '08*.
- [25] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Feives, Steven Glaser, and Martin Turon. 2007. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN '07*.
- [26] K. Langendoen, A. Baggio, and O. Visser. 2006. Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture. In *IPDPS '06*.
- [27] Qinghua Li, Guangjie Li, Wookbong Lee, Moon-il Lee, David Mazzaresse, Bruno Clerckx, and Zexian Li. 2010. MIMO techniques in WiMAX and LTE: A feature overview. *IEEE Commun. Mag* 48, 5 (2010), 86–92.
- [28] Link Labs 2017. (2017). <http://www.link-labs.com/what-is-sigfox/>.
- [29] Dongxin Liu, Zhihao Wu, Fan Wu, Yuan Zhang, and Guihai Chen. 2015. FIWEX: Compressive Sensing Based Cost-Efficient Indoor White Space Exploration. In *MobiHoc '15*.
- [30] LoRa iM880B-L [n. d.]. ([n. d.]). <http://www.wireless-solutions.de/products/radiomodules/im880b-l>.
- [31] LoRa Modem Design Guide 2013. (2013). http://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf.
- [32] LoRaWAN [n. d.]. ([n. d.]). <https://www.lora-alliance.org>.
- [33] LTE Advanced 2017. LTE Advanced Pro. (2017). <https://www.qualcomm.com/invention/technologies/lte/advanced-pro>.
- [34] LTE Standard 2014. THE LTE STANDARD. (2014). <https://www.qualcomm.com/media/documents/files/the-lte-standard.pdf>.
- [35] Yuan Luo, Lin Gao, and Jianwei Huang. 2015. HySIM: A hybrid spectrum and information market for TV white space networks. In *INFOCOM '15*.
- [36] Guoqiang Mao, Bariş Fidan, and Brian D. O. Anderson. 2007. Wireless Sensor Network Localization Techniques. *Computer networks* 51, 10 (2007), 2529–2553.
- [37] Paul Marcelis, Vijay S Rao, and R Venkatesha Prasad. 2017. DaRe: Data Recovery through Application Layer Coding for LoRaWANs. *Proc. ACM/IEEE Internet of Things-Design and Implementation* (2017).
- [38] E. Meshkova, J. Ansari, D. Denkovski, J. Riihijarvi, J. Nasreddine, M. Pavloski, L. Gavrilovska, and P. Mahonen. 2011. Experimental spectrum sensor testbed for constructing indoor radio environmental maps. In *DySPAN '11*.
- [39] A. F. Molisch. 2011. *Wireless Communications (2nd Ed)*. John Wiley and Sons Ltd.
- [40] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl. 2011. SenseLess: A database-driven white spaces network. In *DySPAN '11*.
- [41] R.N. Murty, G. Mainland, I. Rose, A.R. Chowdhury, A. Gosain, J. Bers, and M. Welsh. 2008. CitySense: An Urban-Scale Wireless Sensor Network and Testbed. In *HST '08*.
- [42] NBLoT. 2017. (2017). http://www.3gpp.org/news-events/3gpp-news/1785-nb_10t_complete.
- [43] ngmn [n. d.]. ([n. d.]). <http://www.ngmn.org>.
- [44] E. Oregon and J. Zander. 2010. Short range white space utilization in broadcast systems for indoor environment. In *DySPAN '10*.
- [45] FCC First Order. 2008. (2008). FCC, ET Docket No FCC 08-260, November 2008.
- [46] FCC Second Order. 2010. (2010). FCC, Second Memorandum Opinion and Order, ET Docket No FCC 10-174, September 2010.
- [47] PertoCloud 2017. (2017). <http://petrocloud.com/solutions/oilfield-monitoring/>.
- [48] Ramjee Prasad and Fernando J Velez. 2010. OFDMA WiMAX physical layer. In *WiMAX networks*. Springer, 63–135.
- [49] Ettus Research. 2017. (2017). <http://www.ettus.com/product/details/UB210-KIT>.
- [50] Sid Roberts, Paul Garnett, and Ranveer Chandra. [n. d.]. Connecting Africa Using TV White Spaces: From Research to Real World Deployments. In *LANMAN '15*.
- [51] Abusayed Saifullah, Dolvara Gunatilaka, Paras Tiwari, Mo Sha, Chenyang Lu, Bo Li, Chengjie Wu, and Yixin Chen. 2015. Schedulability analysis under graph routing for WirelessHART networks. In *RTSS '15*.
- [52] Abusayed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. 2016. SNOW: Sensor Network over White Spaces. In *SenSys '16*.
- [53] Abusayed Saifullah, Sriram Sankar, Jie Liu, Chenyang Lu, Bodhi Priyantha, and Ranveer Chandra. 2014. CapNet: A real-time Wireless Management Network for Data Center Power Capping. In *RTSS '14*.
- [54] Scalable Networks [n. d.]. ([n. d.]). <http://web.scalable-networks.com/content/qualnet>.
- [55] SemTech. [n. d.]. LoRa Calculator by Semtech. ([n. d.]). <http://sx1272-lora-calculator.software.informer.com/>.
- [56] Semtech SX1301 [n. d.]. ([n. d.]). <http://www.semtech.com/wireless-rf/rf-transceivers/sx1301/>.
- [57] SIGFOX [n. d.]. ([n. d.]). <http://sigfox.com>.
- [58] Spectrum Bridge 2017. (2017). <http://spectrumbridge.com/tv-white-space/>.
- [59] Chin-Sean Sum, Ming-Tuo Zhou, Liru Lu, R. Funada, F. Kojima, and H. Harada. 2012. IEEE 802.15.4m: The first low rate wireless personal area networks operating in TV white space. In *ICON '12*.
- [60] T. Taher, R. Bacchus, K. Zdunek, and D. Roberson. 2011. Long-term spectral occupancy findings in Chicago. In *DySPAN '11*.
- [61] TinyOS. [n. d.]. ([n. d.]). <http://www.tinyos.net>.
- [62] Understanding FFTs and Windowing 2015. (2015). <http://www.ni.com/white-paper/4844/en/>.
- [63] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Madhusudhan Sudarshan, and Sean Stratman. 2017. FarmBeats: An IoT Platform for Data-Driven Agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, 515–529.
- [64] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. 2016. Mitigating Inter-network Interference in LoRa Networks. *arXiv preprint arXiv:1611.00688* (2016).
- [65] WiMAX [n. d.]. WiMAX. ([n. d.]). <https://en.wikipedia.org/wiki/WiMAX>.
- [66] WirelessHART Specification [n. d.]. ([n. d.]). <http://www.hartcomm2.org>.
- [67] Bei Yin and Joseph R. Cavallaro. 2012. LTE uplink MIMO receiver with low complexity interference cancellation. *Analog Integr Circ Sig Process* 73 (2012), 5443 – 450.
- [68] Xuhang Ying, Jincheng Zhang, Lichao Yan, Guanglin Zhang, Minghua Chen, and Ranveer Chandra. 2013. Exploring Indoor White Spaces in Metropolises. In *MobiCom '13*.
- [69] Jincheng Zhang, Wenjie Zhang, Minghua Chen, and Zhi Wang. 2015. WINET: Indoor white space network design. In *INFOCOM '15*.
- [70] Tan Zhang, Ning Leng, and Suman Banerjee. 2014. A Vehicle-based Measurement Framework for Enhancing Whitespace Spectrum Databases. In *MobiCom '14*.
- [71] Jim Zyren. 2007. Overview of the 3GPP LTE Physical Layer. (2007). <http://www.nxp.com/assets/documents/data/en/white-papers/3GPP-EVOLUTIONWP.pdf>.