SNOW: Sensor Network over White Spaces

Abusayeed Saifullah[†*], Mahbubur Rahman[†*], Dali Ismail[†],
Chenyang Lu[‡], Ranveer Chandra[§], Jie Liu[§]
Department of Computer Science, Missouri University of Science & Technology, Rolla, USA[†]
Cyber-Physical Systems Laboratory, Washington University in St. Louis, St Louis, USA[‡]
Microsoft Research, Redmond, USA[§]

ABSTRACT

Wireless sensor networks (WSNs) face significant scalability challenges due to the proliferation of wide-area wireless monitoring and control systems that require thousands of sensors to be connected over long distances. Due to their short communication range, existing WSN technologies such as those based on IEEE 802.15.4 form many-hop mesh networks complicating the protocol design and network deployment. To address this limitation, we propose a scalable sensor network architecture - called Sensor Network Over White Spaces (SNOW) - by exploiting the TV white spaces. Many WSN applications need low data rate, low power operation, and scalability in terms of geographic areas and the number of nodes. The long communication range of white space radios significantly increases the chances of packet collision at the base station. We achieve scalability and energy efficiency by splitting channels into narrowband orthogonal subcarriers and enabling packet receptions on the subcarriers in parallel with a single radio. The physical layer of SNOW is designed through a distributed implementation of OFDM that enables distinct orthogonal signals from distributed nodes. Its MAC protocol handles subcarrier allocation among the nodes and transmission scheduling. We implement SNOW in GNU radio using USRP devices. Experiments demonstrate that it can correctly decode in less than 0.1ms multiple packets received in parallel at different subcarriers, thus drastically enhancing the scalability of WSN.

CCS Concepts

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

Keywords

White space, Wireless Sensor Network, OFDM

* Co-primary author

SenSys '16, November 14-16, 2016, Stanford, CA, USA

DOI: http://dx.doi.org/10.1145/2994551.2994552

1. INTRODUCTION

Despite the advancement in wireless sensor network (WSN) technology, we still face significant challenges in supporting large-scale and wide-area applications (e.g., urban sensing [61], civil infrastructure monitoring [52, 54], oil field management [23], and precision agriculture [1]). These applications often need thousands of sensors to be connected over long distances. Existing WSN technologies operating in ISM bands such as IEEE 802.15.4 [14], Bluetooth [8], and IEEE 802.11 [13] have short range (e.g., 30-40m for IEEE 802.15.4 in 2.4GHz) that poses a significant limitation in meeting this impending demand. To cover a large area with numerous devices, they form many-hop mesh networks at the expense of energy cost and complexity. To address this limitation, we propose a scalable sensor network architecture - called ***Sensor Network Over White Spaces (SNOW)*** - by designing sensor networks to operate over the TV *white spaces*, which refer to the allocated but unused TV channels.

In a historic ruling in 2008, the Federal Communications Commission (FCC) in the US allowed unlicensed devices to operate on TV white spaces [2]. To learn about unoccupied TV channels at a location, a device needs to either (i) sense the medium before transmitting, or (ii) consult with a cloud-hosted geo-location database, either periodically or every time it moves 100 meters [3]. Similar regulations are being adopted in many countries including Canada, Singapore, and UK. Since TV transmissions are in lower frequencies – VHF and lower UHF (470 to 698MHz) – white spaces have excellent propagation characteristics over long distance. They can easily penetrate obstacles, and hence hold enormous potential for WSN applications that need long transmission range. Compared to the ISM bands used by traditional WSNs, white spaces are less crowded and have wider availability in both rural and urban areas, with rural areas tending to have more [33, 39, 47, 48, 62, 74]. Many wide-area WSNs such as those for monitoring habitat [73], environment [53], volcano [78] are in rural areas, making them perfect users of white spaces. However, to date, the potential of white spaces is mostly being tapped into for wireless broadband access by industry leaders such as Microsoft [20, 65] and Google [29]. Various standards bodies such as IEEE 802.11af [4], IEEE 802.22 [17], and IEEE 802.19 [16] are modifying existing standards to exploit white spaces for broadband access.

The objective of our proposed SNOW architecture is to exploit white spaces for long range, large-scale WSNs. Long range will reduce many WSNs to a single-hop topology that has potential to avoid the complexity, overhead, and latency

associated with multi-hop mesh networks. Many WSN applications need low data rate, low cost nodes, scalability, and energy efficiency. Meeting these requirements in SNOW introduces significant challenges. Besides, long communication range increases the chances of packet collision at the base station as many nodes may simultaneously transmit to it. SNOW achieves scalability and energy efficiency through channel splitting and enabling simultaneous packet receptions at a base station with a single radio. The base station has a single transceiver that uses available wide spectrum from white spaces. The spectrum is split into narrow orthogonal subcarriers whose bandwidth is optimized for scalability, energy efficiency, and reliability. Narrower bands have lower throughput but longer range, and consume less power [37]. Every sensor node transmits on an assigned subcarrier and the nodes can transmit asynchronously. The base station is able to receive at any number of subcarriers simultaneously. The availability of wide white space spectrum will thus allow massive parallel receptions at the base station. Today, all communication paradigms in WSN are point to point, even though convergecast is the most common scenario. Simultaneous packet receptions at low cost and low energy in SNOW represents a key enabling technology for highly scalable WSN. Enabling such simultaneous receptions at a node is challenging as it requires a novel decoding technique which, to our knowledge, has not been studied before.

In SNOW, we implement concurrent transmissions through a Distributed implementation of Orthogonal Frequency Division Multiplexing (OFDM), called ***D-OFDM***, to enable distinct orthogonal signals from distributed nodes. To extract spectral components from an aggregate OFDM signal, we exploit the Fast Fourier Transformation (FFT) that runs on the entire spectrum of the receiver's radio. A traditional decoding technique would require a strict synchronization among the transmissions if it attempts to extract the symbols from multiple subcarriers using FFT. We address this challenge by designing SNOW as an asynchronous network, where no synchronization among the transmitters is needed. The decoder at the base station extracts information from all subcarriers irrespective of their packets' arrival time offsets. Thus, the nodes transmit on their subcarriers whenever they want. The specific contributions of this paper are:

- The Physical layer (PHY) of SNOW that includes white space spectrum splitting into narrowband orthogonal subcarriers and a demodulator design for simultaneous packet receptions; It can decode packets from any number of subcarriers in parallel without increasing the demodulation time complexity. The demodulator also allows to exploit fragmented spectrum.
- The Media Access Control (MAC) protocol for SNOW that handles subcarrier allocation among the nodes and their transmission scheduling.
- Implementation of SNOW in GNU radio using Universal Software Radio Peripheral (USRP) devices; Our experiments show that it can decode in less than 0.1ms all packets simultaneously received at different subcarriers, thus drastically enhancing WSN scalability.

In the rest of the paper, Section 2 outlines the background. Section 3 describes the SNOW architecture. Section 4 presents the PHY of SNOW. Section 5 presents the MAC protocol. Sections 6, 7, and 8 present the implementation, experiments, and simulations, respectively. Section 9 compares SNOW against the upcoming Low-Power Wide-Area Network (LPWAN) technologies. Section 10 overviews related work. Section 11 concludes the paper.

2. BACKGROUND AND MOTIVATION

A WSN is a network of sensors that deliver their data to a base station. It has myriads of applications such as process management [66, 56], data center management [67], and monitoring of habitat [73], environment [53], volcano [78], and civil infrastructure [52]. Many WSNs are characterized by a dense and large number of nodes, small packets, low data rate, low power, and low cost. The nodes are typically battery powered. Thus, scalability and energy are the key concerns in WSN design. Currently, IEEE 802.15.4 is a prominent standard for WSN that operates at 2.4GHz with a bit rate of 250kbps, a communication range of 30-40m at 0dBm, and a maximum packet size of 128 bytes (maximum 104 bytes payload). In this section, we explain the advantages and challenges of adopting white space in WSN.

2.1 White Spaces Characteristics for WSN

Long transmission range. Due to lower frequency, white space radios have very long communication range. Previous [33] as well as our study in this paper have shown their communication range to be of several kilometers. Time synchronization, a critical requirement in many WSN applications, incurs considerable overhead in large-scale and multi-hop deployments which can be avoided in a single-hop structure. Single hop in turn results in shorter end-to-end communication latency by avoiding multi-hop routing.

Obstacle penetration. Wireless communication in 5/2.4GHz band is more susceptible to obstacles. Hence, for example, WirelessHART networks in process monitoring adopt high redundancy where a packet is transmitted multiple times through multiple paths, hindering their scalability [30]. In contrast, lower frequencies of white space allow propagation with negligible signal decay through obstacles.

Many WSN applications need to collect data from sensors spread over a large geographic area. For example, ZebraNet tracks zebras in $200{,}000m^2$ [49]. It lacks continuous connectivity due to the short communication range, and is managed through a delay-tolerant network which cannot deliver information in real time. Also, with the growing applications, industrial process management networks such as WirelessHART networks need to scale up to tens of thousands of nodes [31]. A WirelessHART network relies on global time synchronization and central management that limits network scalability [68]. Having long communication range, white spaces can greatly simplify such wide-area applications.

2.2 Challenge and Approach

WSN characteristics and requirements for scalability and energy efficiency pose unique challenges to adopt white spaces. To achieve energy efficiency, many WSNs try to reduce the idle listening time, employing techniques like low power listening [69] or receiver initiated MAC [71]. However, both cases require one side of the link to send extremely long preambles. Blindly applying existing WSN MAC designs in long communication range will cause most nodes to wake up

unintentionally. Besides, long communication range significantly increases the chances of packet collision.

SNOW achieves scalability and energy efficiency through splitting channels into narrowband orthogonal subcarriers and enabling multiple sensors to transmit simultaneously to the base station with a single radio. Today, all communication paradigms in WSN (and at large) are point to point, even though convergecast is the most common scenario. An n-to-1 convergecast is achieved through n 1-to-1 links. Simultaneous packet receptions at low cost and low energy in SNOW represents a key and novel enabling technology for highly scalable WSN. Such simultaneous receptions at a node is challenging as it requires a novel decoding technique. Our design is based on a distributed implementation of OFDM and we exploit FFT to extract information from all subcarriers. A traditional decoding technique would require that the i-th symbols from all subcarriers be in the same FFT window, requiring strict time synchronization among the transmitting nodes which is difficult for commercially available hardware. We design SNOW as an asynchronous network, where no time synchronization is needed. The decoder can extract information from any number of subcarriers carrying packets irrespective of their packets' arrival time offsets.

3. SNOW ARCHITECTURE

Our proposed SNOW architecture is a WSN with a single base station (BS) and a set of sensor nodes, each equipped with a single half-duplex white space radio. Due to long communication range, all sensor nodes are within a single hop of the BS, and vice versa. We observed in experiment that a node's communication range can be over 1.5km at low transmission power (e.g., 0 dBm). The BS is line-powered, Internet-connected, and powerful. The sensor nodes are power constrained and not directly connected to the Internet.

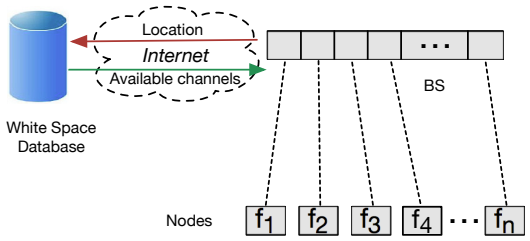


Figure 1: System architecture

The BS uses a wide channel for reception which is split into subcarriers, each of equal spectrum width (bandwidth). Each node is assigned one subcarrier on which it transmits to the BS. Subcarrier allocation to nodes is handled in the MAC protocol. We use the IEEE 802.15.4 [14] packet structure. For integrity check, the senders add cyclic redundancy check (CRC) at the end of each packet. For better energy efficiency, the network does not employ any carrier sensing, RTS/CTS, frame acknowledgment (ACK), or time synchronization protocol. We leave most complexities at the BS and keep the other nodes very simple and energy-efficient. For simplicity, sensors do not do spectrum sensing or cloud access. The BS determines white spaces by accessing a cloud-hosted database through the Internet. We assume that it knows the locations of the nodes either through manual configuration or through some existing WSN localization technique [58]. The BS thus selects white space channels that are available at its own location and at the locations of all other nodes. Figure 1 shows the system architecture of SNOW.

4. SNOW PHY DESIGN

For scalability and energy efficiency, we design the PHY based on channel splitting and by enabling simultaneous packet receptions on different subcarriers at the BS with a single radio. This is done through D-OFDM which is a distributed implementation of OFDM to enable distinct orthogonal signals from distributed sources. We first explain how D-OFDM is realized in SNOW. Then we explain how each subcarrier is modulated for data encoding and how the BS demodulates from multiple subcarriers simultaneously.

4.1 Adopting D-OFDM in SNOW

OFDM is a frequency-division multiplexing (FDM) scheme for digital multi-carrier modulation that uses a large number of closely spaced orthogonal subcarrier signals to carry data on multiple parallel data streams. The key aspect in OFDM is maintaining carrier 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. Therefore, two subcarriers at center frequencies f_i and f_j, $f_i \neq f_j$, are orthogonal when over time T [38]:

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

The orthogonal subcarriers can be **overlapping**, thus increasing the spectral efficiency. The guardbands that were necessary to allow individual demodulation of subcarriers in an FDM system would no longer be necessary. As long as orthogonality is maintained, it is still possible to recover the individual subcarriers' signals despite their overlapping spectrums. In OFDM modulation, the subcarrier frequency f_i, $i = 1, 2, \cdots$, is defined as $f_i = i\Delta f$, where Δf is the *subcarrier spacing*, T is one symbol period and Δf is set to $\frac{1}{T}$ for optimal effectiveness. When there are n' subcarrier center frequencies, $\Delta f = \frac{W}{n'} = \frac{1}{n'T}$ with W being the entire bandwidth. The number of usable subcarriers may be less than n' due to the unavailability of side band at the first/last subcarrier. For example, using one TV channel (6MHz) between 547 - 553MHz, if we want each subcarrier of 400kHz bandwidth, we have $n' = 30$, $\Delta f = 200$kHz. The relative subcarrier frequencies become 200, 400, 600, $\cdots$, 1000kHz. Thus, there will be 29 orthogonal subcarriers with center frequencies 547.2, 547.4, $\cdots$, 552.8MHz from this one TV channel.

While the traditional OFDM is used between a single sender and a single receiver for increased data rate or to increase the symbol duration for enhanced reliability, we adopt D-OFDM in SNOW by assigning the orthogonal subcarriers to different nodes. Each node transmits on the assigned subcarrier. Thus the nodes that are assigned different subcarriers can transmit simultaneously. These component sinusoids form an aggregate time domain signal as follows.

$$X(t) = \sum_{k=0}^{n'-1} x(k)\sin(\frac{2\pi kt}{n'}) - j\sum_{k=0}^{n'-1} x(k)\cos(\frac{2\pi kt}{n'}). \quad (1)$$

where $X(t)$ is the value of the signal at time t which is composed of frequencies denoted by $(2\pi kt/n')$, k is the index of frequency over n' spectral components that divides the available bandwidth with equal spacing and $x(k)$ gives the value of the spectrum at k-th frequency. As seen in Equation (1), any part of the spectrum can be recovered by suitably selecting the spectral coefficients $x(k)$. This is the key principle we adopt in decoding parallel receptions at the BS. We design the demodulator for the receiver of this signal in a way so that no synchronization among these transmitters is needed.

4.2 Modulation Technique

The method for extracting information from multiple subcarriers from an aggregate D-OFDM signal depends on the modulation technique used for encoding the baseband in the carrier signal. We design the PHY of SNOW based on amplitude-shift-keying (ASK) modulation that was adopted in the IEEE 802.15.4 (2006) standard at 868/915MHz [14].

ASK is a form of Amplitude Modulation (AM) that represents digital data as variations in the amplitude of a carrier wave. In an ASK system, the binary symbol 1 is represented by transmitting a fixed-amplitude carrier wave and fixed frequency for a duration of T seconds, where T is the symbol duration. If the signal value is 1 then the carrier signal will be transmitted; otherwise, a signal value of 0 will be transmitted. Every symbol thus carriers one bit. We use the simplest and most common form of ASK, called on-off keying (OOK), in which the presence of a carrier wave indicates a binary one and its absence indicates a binary zero. While AM is not as noise-immune as Frequency Modulation (FM) or Phase Modulation (PM) because the amplitude of the signal can be affected by many factors (interference, noise, distortion) resulting in bit errors, this limitation can be mitigated through bit spreading techniques [10].

The simplicity of AM receiver design is a key advantage of AM over FM and PM [75]. Compared to AM, PM needs more complex receiving hardware. Low bandwidth efficiency is another limitation of PM. The easiest method for AM receiver is to use a simple diode detector. AM transmitter also is simple and cheap as no specialized components are needed. Such a simple circuitry consumes less energy. FM needs comparatively wider bandwidth to handle frequency leakage while AM needs narrower bandwidth as it can be implemented by just making the carrier signal present or absent. Narrower bandwidth in turn consumes much less energy as transmission (Tx) energy is consumed by every Hz of bandwidth. At the same Tx power, the transmitter with narrower bandwidth has longer range. As AM needs narrower bandwidth, the available white space spectrum can be split into a larger number of subcarriers, enhancing SNOW scalability. Thus, there are trade-offs between AM and FM or PM as a modulation technique which is not the focus of this paper.

For robustness in decoding, the modulation maps each bit to a r-bit sequence that simply repeats the bit r times using bit spreading technique. We discuss the choice of parameter r in the following subsection. At the transmitter, bits are mapped to symbols, and then a complex signal is generated. There are only two types of symbols, each consisting of one bit, the signal level above a threshold representing '1' and '0' otherwise. Our work can easily be extended to Quadrature Amplitude Modulation (QAM) that encodes data on both I-signal and Q-signal, thereby doubling the bit rate.

4.3 Demodulator Design

The BS receives an analog D-OFDM signal in time domain and converts it to a digital signal and feeds the digital samples into the SNOW demodulator. We now detail the technique for decoding data from multiple subcarriers.

The transmitters transmit on subcarriers whenever they want without coordinating among themselves. The idea for handling such an asynchronous scenario is to allow the BS to receive anytime. Since the BS is line-powered and has no energy constraints, this is always possible. The BS keeps running an FFT algorithm. The key idea in our demodulator design is to apply an FFT as a ***global FFT Algorithm*** on the entire range of the spectrum of the BS, instead of running a separate FFT for each subcarrier. The demodulator starts processing by storing time domain sequential samples of the received aggregate signal into a vector v of size equal to the number of FFT bins. The global FFT (called FFT for simplicity throughout the paper) is performed on vector v. This repeats at every cycle of the baseband signal.

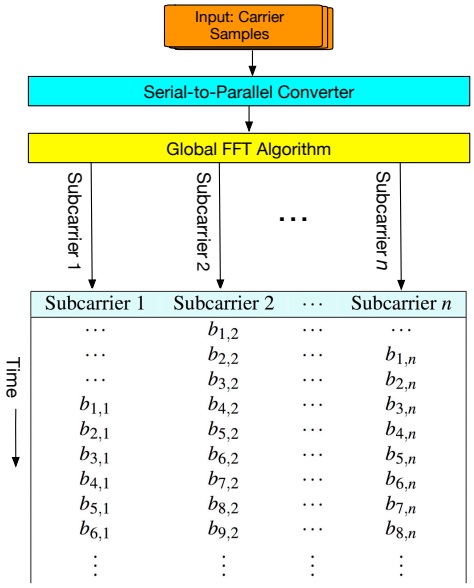


Figure 2: Steps of packet decoding

A workflow showing the various steps for decoding packets from multiple subcarriers in our demodulator is given in Figure 2. A Fourier transform decomposes a time domain signal into a frequency domain representation. The frequency domain represents energy level at each frequency (frequency bins) of that time domain signal. To handle n subcarriers, we apply an m point FFT algorithm, where $m \geq n$, which is a variation of discrete Fourier transform at m frequency bins. Note that the number of subcarriers n depends on the available spectrum, subcarrier spacing, desired bit rate and subcarrier bandwidth which are theoretically explained in Sections 4.1 and 4.4.3, and are experimentally evaluated in Section 7. 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. Each FFT output gives us a set of m values. Each index in that set repre-

sents a single energy level at the corresponding frequency at a time instant. Since our FFT size is fixed no matter how many nodes transmit concurrently, it can decode packets from any number of subcarriers in parallel without increasing the demodulation time complexity. However, the more the number of bins per subcarrier, the cleaner the signal on it.

Handling Spectrum Leakage. FFT algorithm works on a finite set of time domain samples that represent one period of the signal. However, in practice, the captured signal may not be an integer multiple of periods. In that case, finiteness of measured signal results in a truncated waveform. Thus, the end-points become discontinuous and FFT outputs some spectral components that are not in the original signal, letting the energy at one spectral component leak into others. To mitigate the effects of such *spectral leakage* on the neighboring subcarriers, we adopt the *Blackman-Harris windowing* [32]. *Windowing* multiplies a discontinuous time domain records by a finite length window. This window has amplitudes that vary smoothly and gradually towards zero at the edges, minimizing the effects of leakage. Blackman-Harris windowing works for random or mixed signals and gives the best resolution in terms of minimizing spectral leakage.

Packet Decoding. To detect the start of a packet at any subcarrier, the demodulator keeps track of FFT outputs. Since the FFT outputs energy level at each subcarrier, the demodulator applies a threshold to decide whether there is data in the signal. It uses the same threshold to detect preamble bits and the data bits. Once a preamble is detected on a subcarrier, the receiver immediately gets ready to receive subsequent bits of the packet. If the modulation technique spreads one bit into r bits, the demodulator collects samples from r FFT outputs for that subcarrier and then decides whether the actual bit was zero or one. First the packet header is decoded and payload and CRC length is calculated. Then it knows how many data bits it has to receive to decode the packet. 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. The last step in Figure 2 shows the 2D matrix where entry $b_{i,j}$ represents i-th bit of j-th subcarrier. The demodulator starts storing in a column only if a preamble is detected in the corresponding subcarrier. Hence, it stores data and CRC bits for every transmitter when needed. On each subcarrier, when the number of bits stored in the corresponding column of the 2D matrix equals the length of data and CRC bits, we check the CRC and test the validity of reception, and then continue the same process.

Handling Fragmented Spectrum. An added advantage of our design is that it allows to use fragmented spectrum. Namely, if we cannot find consecutive white space channels when we need more spectrum, we may use non-consecutive white spaces. The global FFT is 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.

4.4 Design Parameters

We now discuss some design parameters that play key roles in SNOW operation. We perform signal processing at digitized baseband samples. Those samples are fixed-point precision once converted from the analog domain. For baseband processing, the true measured values in units of current or voltage are not important because those values depend on number representation in the design and the dynamic range of the ADC and prior analog components. Thus, the units of all our parameters are to be interpreted as absolute values.

4.4.1 Threshold Selection

In our decoding, threshold selection on signal strength is a key design parameter to extract information from the received signal. Specifically, the received signal value above the threshold will be considered bit '1', and '0' otherwise. We consider the average signal power to decide the threshold. The average Received Signal Strength (RSS) is estimated using the formula $\sum_{i=1}^{M} \sqrt{I^2 + Q^2}$, where the I and Q are the in-phase and quadrature components, respectively, of the signal, and M is the averaging number of samples.

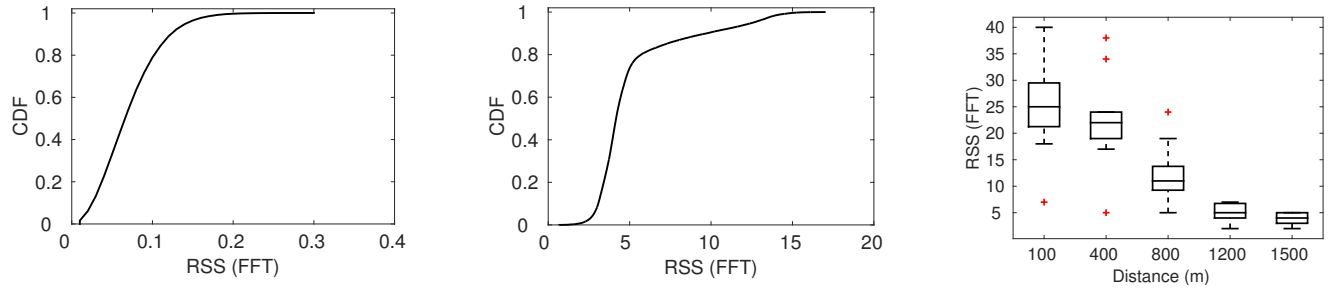
For selecting the threshold, we observe the variability of the spectrum over a period of time and the effect on the RSS at the receiver. We analyzed the spectrum and collected the spectrum data from radio front-ends for a period of 3 weeks. In the receiver we gathered the RSS values for over 50000 samples for the whole duration of the experiment in indoor and outdoor environment that showed us that we can select a steady threshold for packet decoding. Figure 3(a) shows the cumulative distribution function (CDF) of the magnitudes of 50,000 samples for '0' transmission. As it shows, all 100% samples have magnitudes below 0.4 FFT magnitudes. Figure 3(b) shows the CDF of the RSS values for 50000 samples at the same receiver for '1' transmission. In more than 80% cases, the magnitude is above 4.5 while in more than 98.5% cases, it is above 3, implying that we can set a threshold of 3. Figure 3(c) shows the distribution in boxplot for '1' transmission over various distances. At each distance, the boxplot shows the distribution of 5000 samples. All RSS magnitudes including the outliers in all cases are above 5 FFT magnitudes. The results show that a threshold between 0.4 and 5 can effectively distinguish between 1 and 0.

4.4.2 Bit Spreading

Bit spreading is a well-known technique for reducing bit errors in noisy environments by robustly discerning the expected signal and the noise in many wireless technologies such as IEEE 802.15.4 [14] and IEEE 802.11b [13]. In IEEE 802.15.4 based hardware, the Direct Sequence Spread Spectrum (DSSS) technique maps the actual data bits to a different set of bits called *chip-sequence* whose number of bits is 8 times the number of actual data bits [10]. Similarly, in our design using ASK modulation, we adopt bit spreading where every data bit is spread over 8 bits. Our experimental results (Section 7) confirm that this bit spreading helps decode packets correctly even in various noisy conditions.

4.4.3 Packet size, Subcarrier Width, and Bit Rate

We use 28 bytes payload along with 12 bytes header totaling 40-byte as our default packet size in our experiment. TelosB mote [27], a representative WSN mote based on IEEE 802.15.4, uses a default payload of 28 bytes in TinyOS [28]. All results shown in the paper are based on 40-byte packets. The subcarrier bandwidth is another important parameter to decide. The maximum transmission bit rate C



(a) CDF of RSS magnitudes for '0' transmission

(b) CDF of RSS magnitudes for '1' transmission

(c) Distribution of RSS magnitudes over distances

Figure 3: Threshold behavior

of an AWGN channel of bandwidth B based on Shannon-Hartley Theorem is given by $C = B\log_2(1+S/N)$, where S is the signal power and N is the noise power. The ratio S/N is called *Signal to Noise Ratio (SNR)*. The 802.15.4 specification for lower frequency band, e.g., 430-434MHz band (IEEE 802.15.4c [15]), has a bit rate of 50kbps. We also aim to achieve a bit rate of 50kbps. We consider a minimum value of 3dB for SNR in decoding. Taking into account the bit spreading, 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. Based on Nyquist Theorem, $C = 2B\log_2 2^k$ where 2^k is the number of signal levels needed to support bit rate C for a noiseless channel, a modulation technique that uses 2 signal levels can support $50*8$kbps bit rate for a noiseless channel of bandwidth 200kHz. Since ASK modulation uses 2 signal levels, it is theoretically sufficient for this bit rate and bandwidth under no noise. However, to support this bit rate under noise in practical scenarios we determine a required bandwidth of 400kHz through exhaustive experiments in Section 7.2.

5. MAC PROTOCOL FOR SNOW

The MAC protocol operates in two phases - one phase for upward communication (i.e., the nodes transmit to the BS) of duration t_u and the other for downward communication (i.e., the BS transmits to the nodes) of duration t_d, where $t_u \gg t_d$.

The BS first adopts a greedy approach to select the widest free spectrum in available white spaces. If it needs even wider spectrum it can also use the neighboring white spaces in addition to this widest one, thus using fragmented spectrum. For simplicity of presentation, we consider a single (widest) fragment of spectrum. This spectrum is split into n overlapping orthogonal subcarriers, each of equal width. Each node is then assigned one subcarrier. We first explain the case where the number of nodes $N' \leq n$), thus allowing each node to be assigned a unique subcarrier. We denote the subcarrier assigned to node i, $1 \leq i \leq N'$, by f_i. The BS also chooses a control subcarrier denoted by f_c. This channel is used for control operations during the downward communications. Initially and in the downward phase all nodes switch to f_c. The network starts with a downward control command where the BS assigns the subcarriers to the nodes.

The upward communication phase starts right after the BS notifies all the nodes their assigned subcarriers. The BS informs the nodes that the next t_u seconds will be for upward communication. In this way, the nodes do not need to have synchronized absolute times. The BS switches to the entire spectrum and remains in receive mode. In this phase, all nodes asynchronously transmit their data to the BS on the respective subcarriers. After t_u seconds, each node switches to control subcarrier f_c and remains in receive mode for the downward phase, and remains so until it receives a control command from the BS. The BS now switches to f_c and broadcasts control command. This same process repeats.

When the number of nodes $N' > n$, the nodes are grouped, each group having n nodes except the last group that gets (N' mod n) nodes when (N' mod n) $\neq 0$. Every node in a group is assigned a unique subcarrier so that all nodes in the group can transmit together. The BS, in a downward phase, asks a group to transmit their messages in the next upward phase. The next group can be selected in round robin. Thus, the nodes can safety sleep and duty cycle. In upward phase, a node can transmit its own packets and then immediately go to sleep till the end of the upward phase if it has no more data. In downward phase, the node must stay awake to receive any packets from the BS. We can reduce energy consumption further by having the BS notify the nodes in the first downward packet whether it will send more packets in the same phase.

Spectrum and network dynamics are handled through the downward phase. If the spectrum availability changes, then the new channel assignment is informed in the downward phase. The network uses redundant control channels so that if one control channel becomes noisy or unavailable, it can switch to another. If a new node joins the network, it can use the control channel to communicate with the BS. When it detects signals in the control channel, it waits until the channel becomes idle and transmits its ID and location (assumed to be known) 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.

Since we do not use per packet ACK, a node can proactively repeat a transmission γ times for enhanced reliability. The BS can send to the nodes an aggregate ACK to the nodes, e.g., by sending total received packets from a node in the last cycle based on which a node can decide a value of γ.

6. SNOW IMPLEMENTATION

We have implemented SNOW on USRP devices using GNU Radio. GNU Radio is a toolkit for implementing software-

defined radios and signal processing [12]. USRP is a software-defined radio platform with RF front-ends to transmit and receive in a specified frequency [11]. We have 6 sets of USRP B210 devices for experiment, 5 of which are used as SNOW nodes and one as the BS. On the transmitter (Tx) side, packets are generated in IEEE 802.15.4 structure. We represent the preamble and the packet (data, CRC) using a default GNU radio vector. The vector is then sent to the GNU radio repeat block, which performs bit spreading by repeating each bit 8 times. This baseband signal is then modulated with the carrier frequency. For the BS to receive on multiple subcarriers, we implement the decoder using a 64-point FFT. The decoder incorporates serial-to-parallel converter, FFT, parallel-to-serial converter, and signal processing. We do not need FFT size larger than 64-point because of the limited number of devices we have (as every subcarrier already corresponds to multiple FFT bins). Large-scale implementation is done through simulations in QualNet [24].

Parameter	Value
Frequency Band	547 – 553MHz
Orthogonal Frequencies	549.6, 549.8, 550.0, 550.2, 550.4, 550.6MHz
Tx Power	0dBm
Receive Sensitivity	-85dBm
Tx Bandwidth	400kHz
Rx Bandwidth	6MHz
Packet Size	40 bytes
SNR	6dB
Distance	Indoor: 100m Outdoor: 1.5km

Table 1: Default parameter settings

7. EXPERIMENTS

7.1 Setup

We perform experiments using the SNOW implementation on USRP devices in both indoor and outdoor environments. Figure 4(a) shows outdoor node positions for the longest distance we have tested in the City of Rolla. Figure 4(b) shows the positions of the nodes and the BS in the Computer Science building at Missouri University of Science & Technology. It shows 5 different positions (only the positions, not the actual number of nodes) where the nodes were placed in various experiments. We fixed the antenna height at approximately 5 ft above the ground. We experimented in the band between 547MHz and 553MHz that was a fraction of white spaces in the experimental locale. We define ***Correctly Decoding Rate (CDR)*** as the percentage of packets that are correctly decoded at a receiver (Rx) among the transmitted ones. CDR is used to measure the decoding performance of SNOW. We first present the results on determining the subcarriers. Then we present the results running the MAC protocol. Unless stated otherwise, Table 1 shows the default parameter settings for all of the experiments.

7.2 Subcarrier Determination

We perform experiments to determine how to split a wide spectrum into narrowband subcarriers. Narrower bands have lower throughput but they have longer range, are more resilient to multipath effects, and consume less power [37].

(a) Outdoor node locations in the City of Rolla

(b) Node positions shown on the CS building floor plan

Figure 4: Node positions in experiments

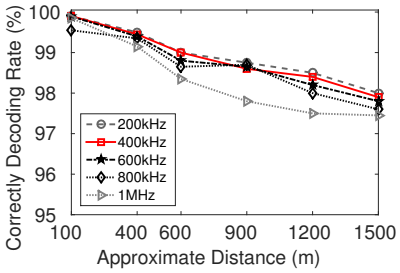
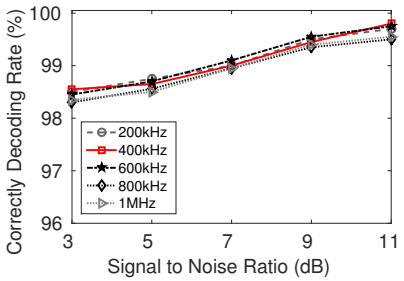


Figure 5: Reliability over long distances (outdoor)

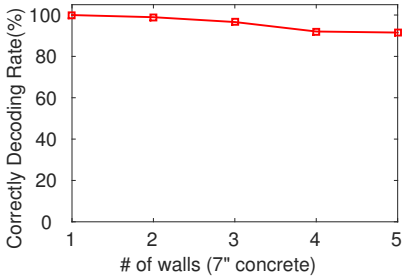
Therefore, we first determine through experiments a feasible bandwidth that is narrow but is sufficient to provide the desired bit rate and to carry WSN packets. In practice, the devices such as TelosB [27] based on the IEEE 802.15.4 standard have a default payload size of 28 bytes in TinyOS [28] which is sufficient to carry WSN data. Therefore, first we set a packet size of 40 bytes of which 28 bytes is payload and 12 bytes is header. We also aim to achieve at least 50kbps bit rate as discussed before. These experiments are performed between two nodes: one node as Tx and the BS as Rx.

7.2.1 *Feasibility of Different Bandwidths over Distances and Obstacles*

Outdoor. We tested in outdoor environments with subcarriers of bandwidths 200kHz, 400kHz, 600kHz, 800kHz, and 1MHz in the band 550 - 551MHz using 0dBm Tx power (which is our default Tx power). Considering 10,000 consecutive packet transmissions, Figure 5 shows we have CDR over 97% for each bandwidth when the receiver is up to 1.5km from the transmitter. As expected, at the same Tx power, the narrower bandwidth has better performance over long distances. While we achieve reliability using 200kHz bandwidth (that was the required theoretical bandwidth as



(a) Reliability at various SNR



(b) Propagation through walls

Figure 6: Link level experiment over obstacles (indoor)

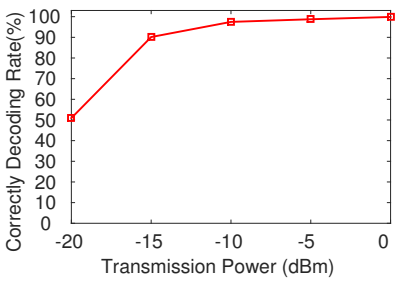


Figure 7: Reliability vs Tx power

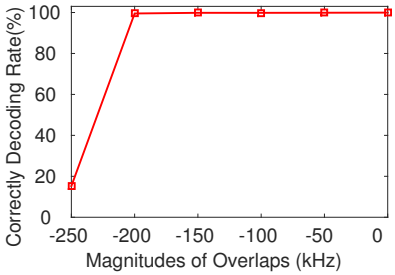


Figure 8: Reliability vs magnitudes of subcarrier overlap

we analyzed in Section 4.3), the bit rate becomes much less than 50kbps. In contrast, when we use 400kHz, we can achieve an effective bit rate of at least 50kbps (8*50kbps in the medium considering spread bits) making 400kHz as our desired subcarrier bandwidth. These results also verify that 40 bytes is a feasible packet size for this bandwidth.

Indoor. We now perform the indoor experiments. Figure 4(b) shows different positions of the transmitter while the receiver is placed in a fixed position. Considering 10,000 consecutive packet transmissions, Figure 6(a) shows the CDR over various SNR conditions for different subcarrier bandwidths. An SNR of 3dB gives a CDR of around 98.5% for all subcarrier bandwidths. As we increase the distances between the BS and the nodes, the SNR changes due to noise, multipath effect, and obstacles. The higher the SNR, the better the CDR. We observe at least 98% CDR on all bandwidths and achieve the desired bit rate when the bandwidth is 400kHz. Based on an experiment using 400kHz bandwidth across obstacles in the same building, Figure 6(b) shows that there is at least 90% CDR when the line of sight is obstructed by up to 5 walls (each 7″ concrete). This shows feasibility of this bandwidth in terms of propagation through obstacles.

7.2.2 *Feasibility under Different Transmission Power*

We now test the feasibility of 400kHz subcarrier bandwidth under different Tx powers. Since USRP devices do not provide any direct mechanism to control Tx power, we perform this experiment by varying the Tx gains at the transmitter to emulate the effect of varying Tx power. Setting a Tx gain of 65dB outputs a Tx power of 0dBm [11]. For 10,000 consecutive packet transmissions in outdoor (Tx and Rx are 1.5km apart), Figure 7 shows the CDR at the receiver under different Tx powers. For Tx power between -15dBm and -10dBm the CDR is at least 97.4%, while for that at 0dBm the CDR is at least 98.1%. The results thus show that when Tx power is not extremely low, 400kHz is a feasible bandwidth.

7.3 Experimenting the SNOW Architecture

We now perform experiments using the complete SNOW architecture under the scenario when multiple nodes transmit to the BS. All of these experiments were done in indoor environments. The node locations are shown in Figure 4(b).

7.3.1 *Overlaps between Orthogonal Subcarriers*

In splitting a wideband radio among multiple orthogonal subcarriers, now we need to analyze the magnitudes of overlaps between the subcarriers. Note that OFDM technology does not require guardband between subcarriers; instead it allows them to be overlapping. We used two subcarriers each of 400kHz bandwidth. Starting with 0 guardband (start of the second subcarrier - end of the first subcarrier), we keep decreasing the value up to the point when the two subcarriers overlap by 50% (representing a guardband of -200kHz).

To evaluate the feasibility of simultaneous reception on overlapping subcarriers, we start transmitting at the two transmitters at the same time. Considering 5,000 consecutive packet transmissions from both of the transmitters, Figure 8 shows a CDR of at least 99.5% at the BS when there is an overlap of 50% or less between these two neighboring subcarriers. While orthogonality allows these overlaps, such a high reliability is achieved not only for orthogonality but also for bit spreading. We observed that there are frequency leakages interfering nearby subcarrier bins, but those were not enough to cause decoding error due to bit spreading. In addition, using multiple bins per subcarrier also helped us reduce the impact of leakage. If we try to move two sub-

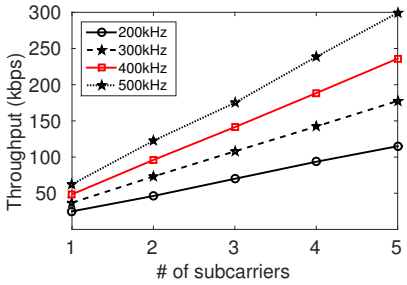


Figure 9: Throughput vs # of subcarriers in SNOW

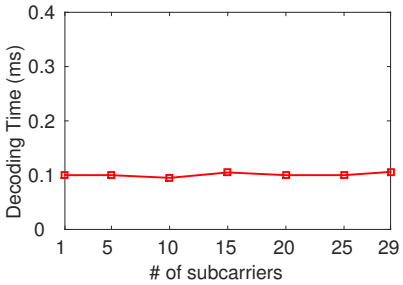


Figure 10: Decoding time vs # of subcarriers

carriers even closer, they affect each other and CDR sharply drops to 5-10%. The experiment shows that the orthogonal subcarriers, each of 400kHz bandwidth, can safely overlap up to 50% with the neighboring ones, thereby yielding high spectrum efficiency (a key purpose of OFDM).

7.3.2 Network Performance

We evaluate some key features of SNOW. First, its *achievable throughput* (total bits received per second at the BS) can be at least n times that of any traditional wireless networks, both having the same link capacity (bit rate on the link) where n is the number of subcarriers. This is because SNOW can receive from n nodes simultaneously. Second, as SNOW runs a single FFT with the same number of bins irrespective of the number of simultaneous transmitters, the time required to demodulate n simultaneous packets is equal to the time needed for decoding a single packet. Now we test these features in experiments. We also evaluate SNOW in terms of energy consumption and network latency.

Throughput:

First we observe the throughput under various number of subcarriers up to 5. The positions of the BS and 5 nodes (indexed as A, B, C, D, E) are shown in Figure 4(b). Each node transmits 40-byte packets consecutively at their maximum bit rate. Thus the throughput measured at the BS indicates the maximum achievable throughput under this setting. The subcarriers are chosen with 50% overlapping with the neighbor/s. In addition to our chosen 400kHz bandwidth, we also experiment with various bandwidths (200kHz, 300kHz, 500kHz) to see the throughput change. Figure 9 shows the throughput averaged over a time interval of 1 hour. When each subcarrier has a bandwidth of 400kHz, the throughput using one transmitter is at least 50kbps. This throughput at the BS increases linearly as we increase the number of transmitters. This increase happens due to parallel receptions on multiple subcarriers at the BS. Note that under similar settings, a traditional WSN will not observe such increased throughput as its radio can receive only if one transmitter transmits at a time. At wider bandwidth, the throughput in SNOW becomes even higher. Thus when we have small number of nodes (compared to the number of subcarriers) and need high throughput, we can choose wider subcarriers.

Decoding Time:

Since the BS in SNOW can receive n packets concurrently, we measure how much time its demodulator takes to handle multiple transmitters. Within a 6MHz channel, we can accommodate 29 orthogonal subcarriers each of width 400kHz and each overlapping 50% with the neighbor/s. Even though we have only 5 USRP transmitters, we can calculate the decoding time for all 29 subcarriers. To do this, we simply assume other 24 transmitters are sending packets containing all zero bits. Theoretically, decoding time for any number of subcarriers should be constant as the FFT algorithm runs with the same number of bins every time. However, assuming 1 to 29 transmitters, we run separate experiments for each number of transmitters (subcarriers) for 7 minutes, and record the worst case time needed for decoding packets. For all cases, Figure 10 shows that the decoding time remains no greater than 0.1ms. This demonstrates the high scalability of SNOW decoding scheme.

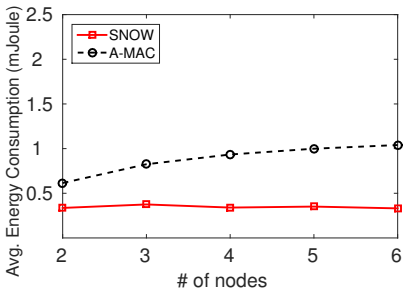
Energy Consumption:

We measure energy consumption in SNOW and compare with that in A-MAC [43] protocol which, to our knowledge, is the state-of-art energy efficient MAC protocol for IEEE 802.15.4 (2.4GHz) based WSNs. A-MAC uses receiver initiated probe to inform the sender to send the packets. Upon receiving the probe the sender sends a hardware generated ACK, followed by the data packet. After receiving the data packet successfully, receiver sends another probe with the ACK bit set. If there are multiple senders, the data packets collide. In that case, the receiver sends a probe containing back-off interval period and backcast channel information.

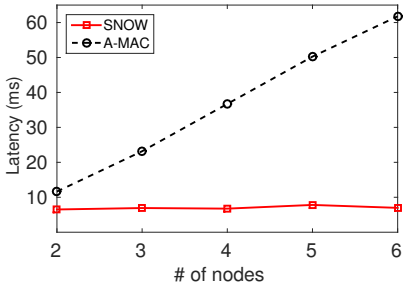
Device mode	**Current Consumption (Supply voltage 3 v)**
Tx	17.5mA
Rx	18.8mA
Idle	0.5mA
Sleep	0.2μA

Table 2: Current Consumption in CC1070

To estimate the energy consumption in SNOW nodes, we place 5 SNOW transmitters each 200m apart from the BS. To make a fair comparison with A-MAC, we place A-MAC nodes 40m apart from each other making a linear multihop network. In both of the networks, each node (except the BS) sends one 40-byte packet every 60 seconds. Since USRP platform does not provide any energy consumption information, we use CC1070 RF transmitter energy model by Texas Instruments [9] to determine approximate energy consumptions in SNOW. This off-the-shelf radio chip has the PHY configuration close to SNOW as it operates in low frequency (402-470 and 804-940MHz) and adopts ASK as one of its modulation techniques. CC1070 energy model is shown in Table 2. In this setup, the BS is always connected



(a) Energy Consumption vs # of nodes



(b) Latency vs # of nodes

Figure 11: Energy consumption and latency

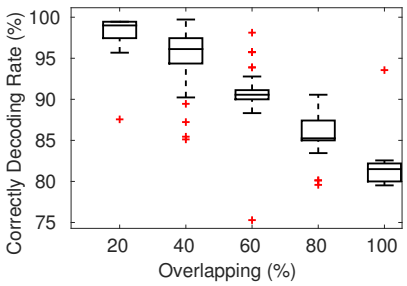


Figure 12: Performance of SNOW under interference

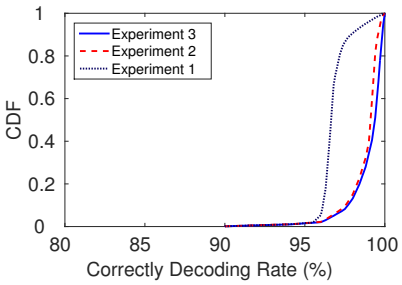


Figure 13: Using fragmented spectrum in SNOW

to a power source and is not considered in energy calculation. We run many rounds of convergecast for one hour. Figure 11(a) shows the average energy consumption in each node per convergecast. Regardless of the number of nodes, a SNOW node consumes 0.34mJoule energy. In contrast, a node under A-MAC consumes on average 0.62mJoule energy when we consider 2 nodes. Average energy consumption on each node in A-MAC increases with the number of nodes. This happens as we increase the number of hops (in the linear topology). Figure 11(a) shows that average energy consumption is 1.04mJoule for 6 nodes in A-MAC while it is almost constant in SNOW. Due to single-hop topology (thanks to long range) and parallel reception at the BS, each node in SNOW consumes less energy on average. This demonstrates the energy efficiency of SNOW over traditional WSN.

Network Latency:

Figure 11(b) shows the comparison of convergecast latency between SNOW and A-MAC with the previous experimental settings. Considering each node has a packet, we measure the latency required to collect all of those packets at the BS. SNOW takes approximately 7ms while A-MAC takes nearly 62ms to collect all the packets in convergecast. It is also noticeable that SNOW needs almost constant time to collect all the packets regardless of the number of nodes as the number of nodes does not exceed the number of subcarriers. Owing to a small network in this experiment (6 nodes), the difference between the latency in A-MAC and that in SNOW cannot be very high. However, for larger networks we will show in simulation that this difference can be very high, demonstrating the scalability of SNOW.

7.3.3 *Performance in the Presence of Interference*

We create interference to see its impact on SNOW performance. We run the upward phase of the MAC protocol where 4 transmitters send packets to the BS concurrently and incessantly. We take another transmitter to act as an interferer. We use the same Tx gain at each transmitter, and place the interferer close (at place A while the legitimate transmitters at places B, C, D, and E in Figure 4(b)) to the BS to have its signal strong at the BS. The interferer operates on different parts of the spectrum of one (of the four) legitimate transmitter, and in every case it uses a timer that fires after every 200ms. At some random time in each of this 200ms window, it transmits a 40-byte packet to the BS. For every magnitude of subcarrier overlapping, we run the experiments for 2 minutes, and measure the CDR at the BS. We perform 50 runs of this experiment and plot the distribution of CDR values in Figure 12. As it shows, with 80% overlap between the subcarriers of a legitimate Tx and the interferer we can decode at least 79% of packets from legitimate Tx in all runs. For 100% overlap, we can decode at least 77% of packets in all runs. This result shows how external interferences can affect SNOW performance. As the figure shows, this impact is less severe or negligible when the interferer's spectrum partially overlaps with the transmitter's subcarrier.

7.3.4 *Performance under Fragmented Spectrum*

An added feature of SNOW is its capability in exploiting fragmented white space spectrum. As primary users may use channels that are far away from each other, white spaces can be largely fragmented. To test the performance of SNOW in fragmented spectrum, we choose different local TV channels such that there are white spaces available on both sides. In this experiment, the BS uses a bandwidth of 8MHz where 6MHz in the middle is occupied by some TV channel. We use two transmitters that act as SNOW nodes and consider three different channels to do three experiments under different fragmented spectrum. Both of the transmitters send

100 consecutive packets and then randomly sleep between 500 to 1000ms. We run this experiment for 2 hours around each channel. In all cases, we run FFT over the entire 8MHz channel and collect data from SNOW nodes only. Under different fragmented spectrum, the SNIR (Signal-to-Noise and Interference Ratio) is different as the TV channels change. Figure 13 shows three sets of experiments on fragmented spectrum, each having different ranges of SNIR condition. In experiment 1, the SNIR varies from 3 to 5dB and SNOW achieves at least 95% CDR in at least 96% cases. In experiment 2, the SNIR varies from 6 to 8dB that results in at least 99% CDR in 90% cases. Experiment 3 with varying SNIR from 9 to 11dB or more shows even better CDR. The results show that SNOW can exploit fragmented spectrum.

8. SIMULATIONS

We evaluate the performance of SNOW for large-scale networks through simulations in QualNet [24]. We evaluate in terms of latency and energy consumption.

8.1 Setup

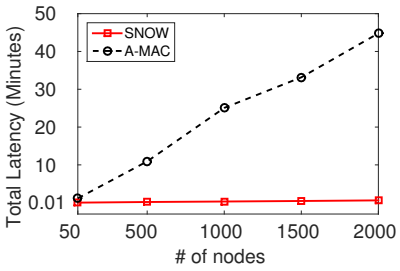
For SNOW, we consider 11MHz spectrum from white space and split into 50 (400kHz each) orthogonal subcarriers each overlapping 50% with the neighbor/s. Putting the BS at the center, we create a star network placing the nodes within 1.5km radius. We generate various numbers of nodes in the network, each in the direct communication with the BS. Since A-MAC is designed for short range WSN (e.g., approx. 40m at 0dBm Tx power), for simulations with A-MAC we place nodes to cover 1.5km radius, making a 38-hop network. In both networks, we perform convergecast. Every node has 100 packets to deliver to the BS. A sleep interval of 100ms is used after a node transmits all of its 100 packets. Each packet is of 40 bytes and is transmitted at 0dBm.

Starting with 50 nodes, we test up to 2000 nodes. We calculate the total latency and the average energy consumption at each node (i.e., the ratio of total energy consumed by all nodes to the number of nodes) to collect all of these 100 packets from all of these nodes at the BS. For SNOW, we assign energy model of CC1070 radio as given in Table 2 to each node. For A-MAC, we assign energy model of CC2420 radio which is roughly similar to that of CC1070 radio.

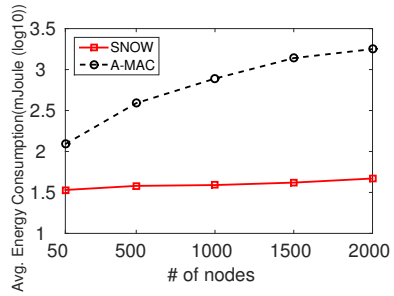
For A-MAC, we run the default TinyOS [28] Collection Tree Protocol [45] with proper configuration wiring [43]. As the network is multi-hop, many nodes also forward packets received from other nodes. All the transmitters keep retrying a packet until they receive a probe with ACK bit set. When we receive at least 90% of all the packets at the BS, we stop data collection for both of the networks.

8.2 Result

Figure 14(a) shows the overall latency for both SNOW and A-MAC for collecting 100 packets from each node at the BS. The latency in A-MAC increases sharply as the number of nodes increases. Up to 50 nodes, SNOW has a total latency of 0.013 minutes as opposed to 1.15 minutes in A-MAC. For 1000 nodes, the A-MAC latency is 25 minutes (vs 0.31 minutes in SNOW) which increases to 45 minutes (vs 0.67 minutes in SNOW) for 2000 nodes. The latency in A-MAC is very high due to collisions, back-off, and probably retransmissions as well. As already acknowledged in [43], A-MAC tends to perform worse in dense neighborhood and high packet delivery scenarios. On the other hand, latency



(a) **Latency for convergecast**



(b) **Avg. energy consumption per node**

Figure 14: Latency and energy consumption in simulation

in SNOW is negligible compared to A-MAC. In SNOW, increasing the number of nodes above 50 increases the overall latency because only 50 nodes can transmit simultaneously.

Figure 14(b) shows average energy consumption on each node when there are various numbers of nodes. We represent the energy information in $\log_{10}$ scale to give a better visibility. For 50-node network, an A-MAC node consumes on average 123.27mJoules for delivering 100 packets compared to 35.2mJoules in SNOW node. For 1000 nodes, these values are 780.12 and 38.33, respectively. For 2000 nodes, these values are 1765.89 and 45.05, respectively. In A-MAC, average energy consumption per node increases sharply as the total number of nodes increases because of higher chances of collisions, back-offs, and retransmissions. As SNOW does not experience collision, its average energy consumption per node increases negligibly with the number of nodes. This justifies the low energy consumption behavior in SNOW.

9. COMPARISON BETWEEN SNOW AND EXISTING LPWAN TECHNOLOGIES

While still in their infancy, LPWAN technologies are gaining momentum in recent years, with multiple competing technologies being offered or under development. The newly certified NB-IoT standard [21] operates over existing cellular networks. NB-IoT and 5G [22] are designed for expensive licensed bands. SIGFOX [25] and LoRa [19] operate in unlicensed ISM band. Their field devices usually need to subscribe to the gateway towers. The radio emitters are required to adopt duty cycled transmission of 1% or 0.1%, depending on the sub-band. Thus they are less suitable for many WSN applications that need real-time requirements or 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 [5] while SNOW can transmit such

a 12-byte message in less than 2ms. LoRa data rates range from 0.3 to 50kbps depending on the bit spreading factor (SF), and allows a user-defined packet size that impacts on Tx range. A high SF enhances reliability but reduces the effective data rate. For example, using 125kHz bandwidth, SFs of 11 and 10 give bit rates of 440bps and 980bps, respectively. Using 125kHz bandwidth and SF of 10, a 10-byte payload packet has an air time of 264.2ms typically [18], which is very large compared to SNOW. SIGFOX and LoRa achieve long ranges using a Tx power up to 20dBm (27dBm for LoRa in USA). SNOW was tested up to 1.5km for which the devices needed a very low Tx power (0dBm or less) which is similar to that achievable in LoRa [36].

For SIGFOX, there exists no publicly available specification or implementation. Hence, an experimental comparison between SNOW and this proprietary technology is beyond our reach at this time. The LoRa specification, designed and patented by Semtech Corporation, has recently been made openly available. Version 1.0 of the LoRaWAN specification was released in June 2015, and is still going through several amendments. While an open source MAC implementation for it was recently released by IBM, it is still going through multiple major updates to be compatible with Semtech modules [6]. It has just been updated to LoRaWAN Specification v1.0.1 in July of 2016 [7]. Thus, even though this standard is promising, the devices and protocols are still under active development. Hence, we leave the experimental comparison with LoRa as a future work. However, we provide some numerical comparison in terms of scalability as follows.

Scalability of SIGFOX/LoRa is achieved assuming extremely low traffic. For example, if a device sends one packet per hour, a LoRaWAN SX1301 gateway using 8 separate radios to exploit 8 channels can handle about 62,500 devices [19]. With its 12-byte message and 140 messages per device per day, one SIGFOX gateway can support 1 million devices [25]. 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. We first consider only upward communication. 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 $4.45m$ million devices. This back-of-envelop calculation indicates SNOW may support significantly more devices than SIGFOX and LoRa. This advantage stems from SNOW's capability to support simultaneous transmissions on multiple subcarriers within a single TV channel.

Another important advantage of SNOW is that it is designed to exploit white spaces which have widely available free spectrum (between 54 and 698MHz in US). In contrast, SIGFOX/LoRa has much less and limited spectrum to utilize (863–870MHz in EU, 902–928MHz in US). The upcoming IEEE 802.15.4m [70] standard aims to exploit white spaces as an extension to the IEEE 802.15.4 standard. Our results can therefore help shape and evolve such standards.

10. RELATED WORK

Several measurement and analytical studies have shown that there exist abundant white spaces in outdoor [62, 48, 74, 47, 39] and indoor [63, 64, 77, 59, 80, 81] environments. Prior work focused on opportunistically forming a single link [35], spectrum sensing [50, 51], and identification of primary users. Later, white spaces were exploited for establishing Wi-Fi like network [33, 83], video-streaming [76], mobile devices [72, 82], dynamic spectrum access [41, 79], and designing a prototype system for spectrum sensing [34, 55]. As spectrum sensing is no longer compulsory, the FCC has recently mandated the use of a geo-location service [26]. The geo-location approach has been widely studied using databases to store white space information for clients query [46, 44, 60, 82, 57]. All of these works consider using white spaces for wireless broadband service. In contrast, we have proposed WSN over white spaces.

Our work is most related to SMACK [42] and WiFi-NC [40]. SMACK [42] was designed for allowing ACK of single-hop broadcast made by an access point. This was done by assigning a subcarrier to each client node that sends an ACK by sending or not sending a tone which is sensed by the access point through energy detection. All such ACKs need to arrive (almost) at the same time - within a window of few microseconds. SMACK is not capable of decoding data from subcarriers and is not designed for handling simultaneous packet reception on multiple subcarriers. WiFi-NC uses a wideband radio as a compound radio that is split into multiple narrowband channels called *radiolets*. Each radiolet is entirely implemented as a separate digital circuit allowing for independent carrier sensing, decoding logic, transmission, and reception of packets in its own narrow channel. Specifically, the transmitter circuit of each radiolet consists of a baseband transmitter, an upsampler, a low pass filter, and a mixer. The receiver circuit of each radiolet consists of a mixer, a low pass filter, a down sampler, and a baseband receiver. Thus the architecture of a WiFi-NC compound radio with m' radiolets is close to that of m' transceivers with low form factor benefits. In contrast, SNOW needs no extra circuitry for any subcarrier. The BS uses a single radio that can receive simultaneously on multiple subcarriers using a single decoding algorithm with no extra hardware or circuit.

11. CONCLUSION AND DISCUSSION

We have designed and implemented SNOW, a scalable and energy-efficient WSN architecture over white spaces. It achieves scalability and energy efficiency through a PHY design that splits channels into narrow band orthogonal subcarriers and enables simultaneous packet receptions with a single radio. SNOW is implemented in GNU radio using USRP devices. Experiments demonstrate that it can decode correctly all simultaneously received packets, thus enabling the scalability for thousands of nodes. In the future, SNOW will be designed based on more robust modulation techniques such as BPSK and QPSK, and with reliable MAC protocols able to handle network and spectrum dynamics, subcarrier hopping, forward error correction, and mobility.

12. ACKNOWLEDGMENTS

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

13. REFERENCES

[1] http://www.nrcs.usda.gov/Internet/FSE_DOCUMENTS/stelprdb1043474.pdf.

[2] FCC, ET Docket No FCC 08-260, November 2008.

[3] FCC, Second Memorandum Opinion and Order, ET Docket No FCC 10-174, September 2010.

[4] http://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11af-white-fi-tv-space.php.

[5] http://www.link-labs.com/what-is-sigfox/.

[6] https://github.com/Lora-net/LoRaMac-node/tree/v3.2.

[7] https://github.com/Lora-net/LoRaMac-node/wiki/LoRaMAC-node-Wiki.

[8] Bluetooth. http://www.bluetooth.com.

[9] CC1070 chip. http://www.ti.com/product/CC1070.

[10] CC2420 chip. http://www.ti.com/product/cc2420.

[11] Ettus Research. http://www.ettus.com/product/details/UB210-KIT.

[12] GNU Radio. http://gnuradio.org.

[13] IEEE 802.11. http://www.ieee802.org/11.

[14] IEEE 802.15.4. http://standards.ieee.org/about/get/802/802.15.html.

[15] IEEE 802.15.4c. https://standards.ieee.org/findstds/standard/802.15.4c-2009.html.

[16] IEEE 802.19. http://www.ieee802.org/19/.

[17] IEEE 802.22. http://www.ieee802.org/22/.

[18] LoRa modem design guide. http://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf.

[19] LoRaWAN. https://www.lora-alliance.org.

[20] Microsoft 4Afrika. http://www.microsoft.com/africa/4afrika/.

[21] NB-IoT. https://www.u-blox.com/en/narrowband-iot-nb-iot.

[22] ngmn. http://www.ngmn.org.

[23] PetroCloud. http://petrocloud.com/solutions/oilfield-monitoring/.

[24] QualNet. http://web.scalable-networks.com/content/qualnet.

[25] SIGFOX. http://sigfox.com.

[26] Spectrum Policy Task Force Reports. http://www.fcc.gov/sptf/reports.html.

[27] TelosB datasheet. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf.

[28] TinyOS Community Forum. http://www.tinyos.net.

[29] TV White Spaces Africa Forum 2013. https://sites.google.com/site/tvwsafrica2013/.

[30] WirelessHART Specification. http://www.hartcomm2.org.

[31] WirelessHART System Engineering Guide. http://www2.emersonprocess.com/siteadmincenter/PM%20Central%20Web%20Documents/EMR_WirelessHART_SysEngGuide.pdf.

[32] Understanding FFTs and windowing, 2015. http://www.ni.com/white-paper/4844/en/.

[33] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with wi-fi like connectivity. In *SIGCOMM '09*.

[34] R. Balamurthi, H. Joshi, C. Nguyen, A. Sadek, S. Shellhammer, and C. Shen. A TV white space spectrum sensing prototype. In *DySpan '11*.

[35] R. E. D. Borth and B. Oberlie. Considerations for successful cognitive radio systems in US TV white space. In *DySpan '08*.

[36] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi. Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, October.

[37] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A case for adapting channel width in wireless networks. In *SIGCOMM '08*.

[38] C.-T. Chen. *System and Signal Analysis*. Thomson, 1988.

[39] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li. Mining spectrum usage data: a large-scale spectrum measurement study. In *MobiCom '09*.

[40] K. Chintalapudi, B. Radunovic, V. Balan, M. Buettener, S. Yerramalli, V. Navda, and R. Ramjee. WiFi-NC: WiFi over narrow channels. In *NSDI '12*.

[41] S. Deb, V. Srinivasan, and R. Maheshwar. Dynamic spectrum access in DTV white spaces: Design rules, architecture and algorithms. In *MobiCom '09*.

[42] A. Dutta, D. Saha, D. Grunwald, and D. Sicker. SMACK: A smart acknowledgment scheme for broadcast messages in wireless networks. In *SIGCOMM '09*.

[43] 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 *SenSys '10*.

[44] X. Feng, J. Zhang, and Q. Zhang. Database-assisted multi-AP network on TV white spaces: Architecture, spectrum allocation and AP discovery. In *DySpan '11*.

[45] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *SenSys '09*.

[46] D. Gurney, G. Buchwald, L. Ecklund, S. Kuffner, and J. Grosspietsch. Geo-location database techniques for incumbent protection in the TV. In *DySpan '08*.

[47] M. Islam, C. Koh, S. Oh, X. Qing, Y. Lai, C. Wang, Y. Liang, B. Toh, F. Chin, G. Tan, and W. Toh. Spectrum survey in Singapore: Occupancy measurements and analyses. In *CrownCom '08*.

[48] V. Jaap, R. Janne, A. Andreas, and M.Petri. UHF white space in Europe: a quantitative study into the potential of the 470-790 MHz band. In *DySpan '11*.

[49] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In *ASPLOS-X '02*.

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

[51] H. Kim and K. G. Shin. In-band spectrum sensing in cognitive radio networks: Energy detection or feature detection? In *MobiCom '08*.

[52] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN '07*.

[53] K. Langendoen, A. Baggio, and O. Visser. Murphy loves potatoes: experiences from a pilot sensor network

deployment in precision agriculture. In *IPDPS '06*.

[54] B. Li, Z. Sun, K. Mechitov, C. Lu, D. Dyke, G. Agha, and B. Spencer. Realistic case studies of wireless structural control. In *ICCPS '13*.

[55] D. Liu, Z. Wu, F. Wu, Y. Zhang, and G. Chen. FIWEX: Compressive sensing based cost-efficient indoor white space exploration. In *MobiHoc '15*.

[56] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen. Real-time wireless sensor-actuator networks for industrial cyber-physical systems. *Proceedings of the IEEE*, 104(5):1013–1024, 2016.

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

[58] G. Mao, B. Fidan, and B. D. O. Anderson. Wireless sensor network localization techniques. *Computer networks*, 51(10):2529–2553, 2007.

[59] E. Meshkova, J. Ansari, D. Denkovski, J. Riihijarvi, J. Nasreddine, M. Pavloski, L. Gavrilovska, and P. Mahonen. Experimental spectrum sensor testbed for constructing indoor radio environmental maps. In *DySpan '11*.

[60] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl. SenseLess: A database-driven white spaces network. In *DySpan '11*.

[61] R. Murty, G. Mainland, I. Rose, A. Chowdhury, A. Gosain, J. Bers, and M. Welsh. CitySense: An urban-scale wireless sensor network and testbed. In *HST '08*.

[62] M. Nekovee. Quantifying the availability of TV white spaces for cognitive radio operation in the UK. In *ICC '09*.

[63] E. Obregon and J. Zander. Short range white space utilization in broadcast systems for indoor environment. In *DySpan '10*.

[64] R. Obregon, L. Shi, J. Ferrer, and J. Zander. Experimental verification of indoor TV white space opportunity prediction model. In *CrownCom '10*.

[65] S. Roberts, P. Garnett, and R. Chandra. Connecting africa using the tv white spaces: From research to real world deployments. In *LANMAN '15*.

[66] A. Saifullah, D. Gunatilaka, P. Tiwari, M. Sha, C. Lu, B. Li, C. Wu, and Y. Chen. Schedulability analysis under graph routing for WirelessHART networks. In *RTSS '15*.

[67] A. Saifullah, S. Sankar, J. Liu, C. Lu, B. Priyantha, and R. Chandra. CapNet: A real-time wireless management network for data center power capping. In *RTSS '14*.

[68] A. Saifullah, C. Wu, P. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen. Near optimal rate selection for wireless control systems. *ACM Transactions on Embedded Computing Systems*, 13(4s):1–25, 2013.

[69] M. Sha, G. Hackmann, and C. Lu. Energy-efficient low power listening for wireless sensor networks in noisy environments. In *IPSN '13*.

[70] C.-S. Sum, M.-T. Zhou, L. Lu, R. Funada, F. Kojima, and H. Harada. IEEE 802.15.4m: The first low rate wireless personal area networks operating in TV white space. In *ICON '12*.

[71] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *SenSys '08*.

[72] S. Sur and X. Zhang. Bridging link power asymmetry in mobile whitespace networks. In *INFOCOM '15*.

[73] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04*.

[74] T. Taher, R. Bacchus, K. Zdunek, and D. Roberson. Long-term spectral occupancy findings in Chicago. In *DySpan '11*.

[75] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[76] X. Wang, J. Chen, A. Dutta, and M. Chiang. Adaptive video streaming over whitespace: SVC for 3-tiered spectrum sharing. In *INFOCOM '15*.

[77] M. Wellens, J. Wu, and P. Mahonen. Evaluation of spectrum occupancy in indoor and outdoor scenario in the context of cognitive radio. In *CrownCom '07*.

[78] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI '06*.

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

[80] X. Ying, J. Zhang, L. Yan, G. Zhang, M. Chen, and R. Chandra. Exploring indoor white spaces in metropolises. In *MobiCom '13*.

[81] J. Zhang, W. Zhang, M. Chen, and Z. Wang. WINET: Indoor white space network design. In *INFOCOM '15*.

[82] T. Zhang, N. Leng, and S. Banerjee. A vehicle-based measurement framework for enhancing whitespace spectrum databases. In *MobiCom '14*.

[83] X. Zhang and E. W. Knightly. WATCH: WiFi in active TV channels. In *MobiHoc '15*.