Learning to Share: Narrowband-Friendly Wideband Networks

Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat
Massachusetts Institute of Technology

ABSTRACT

Wideband technologies in the unlicensed spectrum can satisfy the ever-increasing demands for wireless bandwidth created by emerging rich media applications. The key challenge for such systems, however, is to allow narrowband technologies that share these bands (say, 802.11 a/b/g/n, Zigbee) to achieve their normal performance, without compromising the throughput or range of the wideband network.

This paper presents SWIFT, the first system where high-throughput wideband nodes are shown in a working deployment to coexist with unknown narrowband devices, while forming a network of their own. Prior work avoids narrowband devices by operating below the noise level and limiting itself to a single contiguous unused band. While this achieves coexistence, it sacrifices the throughput and operating distance of the wideband device. In contrast, SWIFT creates high-throughput wireless links by weaving together non-contiguous unused frequency bands that change as narrowband devices enter or leave the environment. This design principle of cognitive aggregation allows SWIFT to achieve coexistence, while operating at normal power, and thereby obtaining higher throughput and greater operating range. We implement SWIFT on a wideband hardware platform, and evaluate it in the presence of 802.11 devices. In comparison to a baseline that coexists with narrowband devices by operating below their noise level, SWIFT is equally narrowband-friendly but achieves $3.6-10.5\times$ higher throughput and $6\times$ greater range.

Categories and Subject Descriptors C.2.2 [**Computer Systems Organization**]: Computer-Communications Networks

General Terms Algorithms, Design, Performance

1 Introduction

Users' desires to share high definition audio and video around the home are driving the need for ever-increasing wireless bandwidth [1, 9], and wideband radios, whose frequency bandwidth spans hundreds of MHz to many GHz, have been proposed as a solution [9, 34, 20]. These radios mainly operate in the unlicensed spectrum, which is populated by a variety of legacy narrowband devices (e.g., 802.11a/b/g, Zigbee), as well as a slew of emerging technologies (e.g., 802.11n). The key problem in operating these wideband systems is to ensure that they neither hinder the performance of narrowband devices in these bands, nor sacrifice their own throughput or operating range. Overcoming this problem requires a network design that achieves high throughput even when interferers continuously exist, a fundamental departure from traditional wireless networks, which are crippled by interference.

This paper presents SWIFT, a **S**plit **W**ideband **I**nterferer **F**riendly **T**echnology that safely coexists with narrowband devices operating in the same frequencies. SWIFT's key feature is cognitive aggregation: the ability to create high-throughput wireless links by weaving together non-contiguous unused frequency bands that change as narrowband devices enter or leave the environment. Our design is motivated by measurement studies [19, 27] showing that, while various wireless technologies exist throughout the spectrum, only a few such technologies are usually operational in a house or small geographic area,[1] and hence a large number of non-contiguous frequency bands are likely to be unused. SWIFT's ability to detect and utilize exactly these unoccupied bands, and compose them to build a single wireless link, allows wideband networks to operate at normal power without affecting narrowband, and delivers on the promise of simultaneously achieving high throughput, operating range, and coexistence.

SWIFT bridges two areas in wireless communications: cognitive radios, and wideband and ultra-wideband design. While there has been a lot of interest in cognitive communication, most proposals have focused on the licensed spectrum [12, 10, 16], where the primary users of the band are known *a priori*, and hence this knowledge may be incorporated into detecting if the band is occupied by the known signal pattern. In contrast, SWIFT focuses on the unlicensed band, where narrowband devices are many, and their signal patterns are unlikely to be known. Further, cognitive proposals attempt to find a single unused band which they may opportunistically use, while SWIFT aggregates the bandwidth of many such bands to maximize throughput. Similarly to cognitive radios, Wideband (WB) and Ultra-wideband (UWB) technologies have to cooperate with existing users of the spectrum. They have, however, tried to bypass the coexistence problem by reducing their transmission power below the noise floor of narrowband devices [34, 29, 4], and limiting themselves to a single contiguous band. While this allows narrowband devices to operate unhindered, it sacrifices the WB device's throughput, operating distance, or both.

To achieve its goal of high throughput, range, and narrowband-friendliness, SWIFT has to address three key challenges:

- *How does SWIFT detect the frequency bands that it must avoid, to allow narrowband devices to operate normally?* In the absence of any information about the narrowband signal, traditional solutions avoid frequency bands that show high narrowband power [10]. This approach uses observed power (or the lack of it) in a band as a proxy for whether interference in this band is detrimental (or irrelevant) to operation of the narrowband device, and is known to have both false positives and false negatives [32]. Instead, SWIFT has a novel *adaptive sensing* technique that exploits common net-

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.

---

[1] The measured average spectrum occupancy is 5.2% [19].

work semantics, by observing that many unlicensed devices react when faced with interference, either at the lower layers [7, 21], or at higher layers [24]. This observation allows SWIFT to directly address the key goal of cognition: identifying frequency bands whose use could interfere with narrowband devices. Thus, SWIFT probes ambiguous frequencies, monitors the change in narrowband power profile, and backs away if it perceives narrowband reaction.

- *How does the PHY layer operate across chunks of non-contiguous frequencies?* The current PHY layer of high-throughput wireless systems assumes a known and contiguous communication band, and breaks down in the presence of narrowband devices. For example, even basic primitives like packet detection can be triggered incorrectly by power from narrowband transmissions. SWIFT introduces a *cognitive PHY* that incorporates cross-layer information from the adaptive sensing subsystem into the basic signal processing algorithms.
- *Given that different nodes might perceive different usable frequencies, how do SWIFT nodes communicate?* Varying proximity to narrowband devices between SWIFT transmitter-receiver pairs may lead to differences in their choice of usable frequency bands. Since state of the art high-throughput wireless systems (e.g. OFDM) communicate across a frequency band by striping the data bits sequentially across sub-frequencies in the band, disagreement in the set of usable sub-frequencies between a sender-receiver pair leads to unknown insertions and deletions in the data stream, which cannot be dealt with by typical error-correcting codes. SWIFT's *in-band consensus scheme* transforms these insertions and deletions into bit errors, which can be dealt with using standard error-correcting techniques, and hence enables communication despite uncertainty in the environment.

We have built SWIFT in a custom wideband radio hardware [20]. Our implementation addresses the major details of computational complexity, storage, and pipelining inherent in building a wideband wireless transceiver and apparent only at the hardware level. We evaluate our design in a testbed of wideband nodes and 802.11 narrowband devices. Our results reveal the following findings.

- SWIFT safely coexists with narrowband devices while simultaneously providing high throughput and good range. In comparison to a baseline system that coexists with narrowband devices by operating below their noise level, SWIFT is as narrowband-friendly, but its throughput is $3.6 - 10.5\times$ higher, and its range is $6\times$ greater.
- Adaptive sensing is effective. As compared to a threshold based approach, which is neither efficient for wideband nor safe for narrowband across all locations, adaptive sensing accurately identifies interfered frequency bands, and provides efficiency while still being safe for narrowband.
- SWIFT nodes can communicate despite disagreement over narrowband spectrum usage and tolerate up to 40% disagreement about the usable frequency bands.

To the best of our knowledge, SWIFT is the first system where wideband nodes are shown in a working deployment to coexist safely with unknown narrowband devices, while forming a network of their own.

2 Related Work

SWIFT brings together research in two threads of wireless communications: wideband systems, and cognitive radios.

(a) Wideband Systems. The last couple of years have seen tremendous successes in the implementation of WB and UWB radios [18, 20, 9, 34]. This work falls in two major categories: low power consumption, low-rate radios for precision location and tracking systems, and high throughput radios for personal area networks and wire replacement in homes and offices [9, 1].

An intrinsic problem for high-throughput wideband radios, however, is coexistence with narrowband devices with which they share the unlicensed bands. Prior work tries to avoid interfering with narrowband devices by transmitting below their noise level [34, 29]. This approach inherently limits the throughput and operating range of the WB radio [34]. Further, in many cases, it fails to achieve its goal of protecting narrowband devices [29, 4]. Mishra *et al.* [28] propose to detect and avoid WiMax operating in the same band as an ultra-wideband device. Their work however is specific to WiMax, and can deal neither with general narrowband devices nor with a dynamic environment. Also, their implementation considers only a wideband sender and does not include a wideband receiver.

While most prior work is focused on a single link and the PHY layer, SWIFT's components span multiple areas, including signal processing, coding, and network protocols, which together successfully address the issue of coexistence with dynamic and unknown narrowband devices.

(b) Cognitive Radios. The realization of the congested spectrum allocation and its inefficient utilization [19, 27] has led to a surge of interest in cognitive communications. Work here has largely focused on detecting unused bands (spectrum sensing) and providing methods for sharing these bands among cognitive radios (spectrum sharing).

Prior work on spectrum sensing focuses on the licensed band, where it is crucial that cognitive secondary users do not interfere with the licensed primary user. The most basic approach involves measuring the energy level in a band. Energy detection is cheap, fast, and requires no knowledge of the characteristics of the signal. However, choosing energy thresholds is not robust across a wide range of SNRs [10]. Though more sophisticated mechanisms such as matched filter detection [10] are more accurate, they require knowledge of the transmitted signal (modulation, packet format, pilots, bandwidth, etc.) and thus work only for known technologies.

Architectures for spectrum sharing fall in two categories: centralized and distributed [10]. Centralized approaches [3, 15, 14] require a controller, such as a base station or spectrum broker, to allocate spectrum to all cognitive users. Distributed approaches [35, 36, 12, 16, 25] have MAC protocols that rely on one or more control channels to coordinate spectrum access.

While our work builds on these prior foundations, it makes three major departures. First, cognitive radios focus on finding a single contiguous unoccupied band, whereas SWIFT weaves together multiple non-contiguous unoccupied bands to create a high-throughput wideband link. Second, SWIFT introduces new spectrum sensing mechanisms that exploit network semantics to strengthen traditional energy based techniques for unknown signals. Third, SWIFT allows communicating nodes to agree on usable frequencies using a fully distributed consensus scheme that requires no control channels.

3 Problem Domain

SWIFT is designed to provide high throughput wireless connectivity for rich media appliances in a home scenario. It operates in the unlicensed spectrum, and is intended to function in the presence of narrowband devices that utilize the same part of the spectrum, and which might persist for long periods, or arrive and depart within minutes or hours, e.g., a laptop utilizing an 802.11 wireless connection.

SWIFT is a cognitive architecture for OFDM wideband radios. We focus on Orthogonal Frequency Division Multiplexing (OFDM) because it has emerged as the technique of choice for the majority of

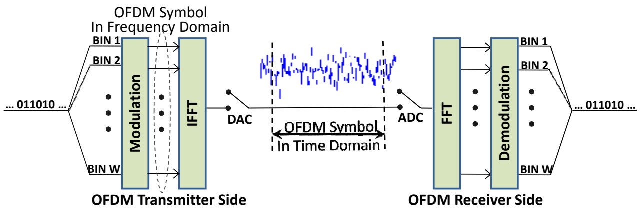


Figure 1: Schematic of an OFDM System

wireless technologies, such as wideband digital communication [20], ultra-wideband [5], 802.11 a/g/n [7, 8] and WiMAX [11]. The rest of our description focuses on single antenna radios, but our ideas are also applicable to wideband MIMO radios, as they too use OFDM [13].

Robust detection of narrowband devices without any knowledge of their signal patterns or other characteristics is impossible [32]. Since it is impractical to assume known signal patterns in the unlicensed band, SWIFT focuses its design on the practical scenarios that could arise in the environment of interest. Specifically, SWIFT addresses situations in which the following constraints apply:

1. It is acceptable to treat narrowband traffic as best effort. Specifically, narrowband devices should continue to experience the same average throughput and loss rate in the presence of wideband nodes as without them, but their requirements are not any more stringent than what is expected from today's wireless LANs.

2. The capacity of the wideband network exceeds its peak traffic. This implies that the medium exhibits frequent idle intervals such that narrowband devices that perform carrier sense are not completely locked out. Sufficient capacity can arguably be obtained by increasing the spectrum width spanned by the wideband radio.

3. Narrowband technologies of interest in this paper react to interference. This reaction can be at lower layers, for example, carrier-sense abstaining from using the medium, or autorate changing modulation schemes, or at the higher layer, for example, TCP backing off on sustained packet loss. Further, these devices are expected to operate at reasonable SNRs (a few dB above the noise floor, e.g. 802.11a/b/g/n). Narrowband devices that operate below or around the noise floor are expected to have their own mechanisms to combat interference, as they need them in such a regime.

4 OFDM Background

This section provides a simplified description of OFDM focused only on issues related to this paper. OFDM divides the used RF bandwidth into many narrow sub-channels, called OFDM bins. Each OFDM bin can be treated independently from other bins, and may use a different modulation (e.g., BPSK, 4-QAM) or transmission power. A data stream is striped into bits, with different numbers of bits assigned to each bin based on its modulation scheme. An assignment of modulated bits to each of the OFDM bins is called an OFDM symbol, see Fig. 1. The frequency representation of the OFDM symbol is converted to a time domain OFDM symbol by using an Inverse Fast Fourier Transform (IFFT) and sent on the medium by the transmitter.

The receiver first determines the exact sample at which the packet starts. It then aligns the time samples on OFDM symbol boundaries, and performs a few basic signal processing tasks like Carrier Frequency Offset (CFO) and channel estimation. Next, the aligned time signal is passed to a Fast Fourier Transform (FFT) module to produce the frequency representation. The data symbols are then converted to their frequency representation, corrected for the channel, and demodulated to retrieve the transmitted data bits.

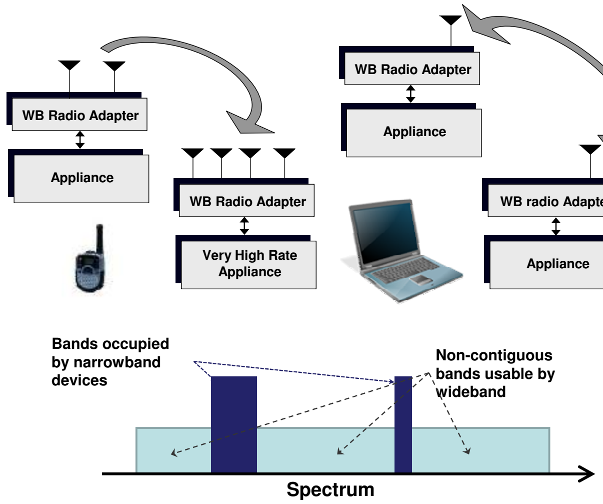


Figure 2: Cognitive Aggregation: While narrowband devices exist (e.g., 802.11 laptop), SWIFT still uses the remaining non-contiguous chunks of spectrum as if they were one wireless link.

5 SWIFT

SWIFT is designed around the concept of *cognitive aggregation*. Similar to the cognitive radio vision, cognitive aggregation is based on detecting narrowband systems and avoiding their frequency bands. Unlike prior cognitive systems, which use only a single contiguous band, cognitive aggregation merges many non-contiguous bands into a single high-throughput communication channel, as shown in Fig. 2. Such a design is critical when using a wide band in the unlicensed spectrum since a wide contiguous unused band typically does not exist. SWIFT implements a cognitive aggregation design by utilizing three key components: (a) a spectrum sensing mechanism based on determining how SWIFT's selection of frequency bands impacts narrowband transmissions, rather than just how the narrowband transmissions look to SWIFT, (b) a cognitive PHY layer that can operate over non-contiguous spectrum bands, and (c) a consensus protocol that allows SWIFT nodes to agree on usable frequency bands despite uncertainty about which bands are occupied by narrowband devices. Below, we explain each of these components in detail.

5.1 Adaptive Spectrum Sensing

SWIFT senders must learn the set of OFDM bins in which they can send while being narrowband-friendly.

5.1.1 How do we detect bins that interfere with narrowband?

Ideally, SWIFT could directly measure how its choice of transmit bins affects a narrowband device. Since this is typically not possible, and given that one does not know the signal details for arbitrary unlicensed narrowband devices, prior cognitive devices passively listen for narrowband devices, and avoid all frequency bins in which they see power above some threshold [10]. This approach essentially uses information about how SWIFT observes the narrowband transmissions to guess how a SWIFT transmission would be observed by the narrowband device. Such an approach is problematic for two reasons.

First, it is difficult to pick a power threshold [32] to precisely identify occupied bins, because the correct value varies with time and proximity to the narrowband device. Fig. 3 illustrates this issue. It shows the power profile of an 802.11a narrowband device operating on channel 52, as observed by two SWIFT nodes at different dis-

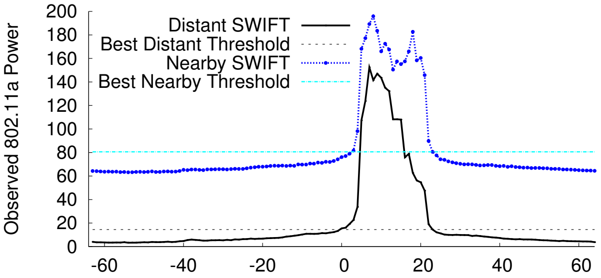


Figure 3: 802.11a Power Profile: The observed power of an 802.11a transmitter at different SWIFT locations is very different, highlighting the difficulty in picking a power threshold that works at all locations.

tances from the 802.11a transmitter. In this scenario, the narrowband device uses bins 3 through 23. Clearly, no single fixed threshold would eliminate exactly the correct set of bins used by the narrowband device at both locations. This problem becomes even worse in the presence of variable power levels among narrowband devices. For example, portable 802.11 devices such as laptops and handheld devices often transmit at power levels well below the maximum in order to conserve their battery, meaning that even though SWIFT's effect on two devices in different locations might be very different, the transmissions from those two devices might be indistinguishable from SWIFT's perspective. Accounting for all this variability requires using a very conservative threshold that wastes many bins.

Second, even if one could identify the exact bins the narrowband device uses for its transmissions, this may not be the correct set of bins to avoid. Since transmitters leak power into bins adjacent to the ones they use, a wideband transmitter might need to avoid bins that are unused by the narrowband device if using them would leak significant power into the narrowband bins. Conversely, a wideband device might be able to use bins that are used by the narrowband device without affecting narrowband operation. This might happen if the narrowband device is far away from the wideband transmitter, or uses highly redundant coding schemes (e.g., Zigbee [37]). Because these effects depend on the distance and receive sensitivity of the narrowband device, it is impossible to account for them without being extremely conservative in the choice of threshold.

The key problem with current solutions is that they use the wideband device's view of the narrowband transmissions in an open loop, as a proxy for how the narrowband device will observe the wideband transmissions. Asymmetric links, and varying transmission powers and receive sensitivities, make this a poor proxy. SWIFT instead uses a technique we call *adaptive sensing*, which closes the loop by taking advantage of the observation that many narrowband devices react in some perceivable way if wideband transmissions disrupt their transmissions. In particular, a large class of narrowband technologies in the unlicensed spectrum reacts to interference, either at lower layers (e.g., carrier-sense and autorate) or higher layers (e.g., TCP or end-user backoff). Intuitively, SWIFT pokes the narrowband device by putting power in ambiguous bins, notes any changes in the narrowband power profile, and backs away if such a reaction is observed.

Note that our goal with adaptive sensing is not to use narrowband bins during short gaps in narrowband transmissions; rather, we design it to immediately relinquish bins that it suspects of being used by narrowband devices, and reuse them only when confident that the narrowband devices have disappeared for several minutes.

5.1.2 Detecting Narrowband Reaction

SWIFT continuously senses the medium whenever it is not sending or receiving a packet. It converts the incoming time signal to the frequency domain using an FFT, and then calculates the current power in each bin. These power measurements are used both to detect the existence of a narrowband device, and to identify whether the narrowband device has reacted to the wideband device.

SWIFT detects the presence of a narrowband device in a bin, by comparing the power in that bin to the *noise floor*. SWIFT computes the noise floor by taking advantage of its wide band. Since it is highly unlikely that narrowband devices are simultaneously present in all bins, SWIFT just computes the minimum power across all bins and averages it over time to estimate the noise floor. Before SWIFT runs its adaptive sensing algorithm to choose the correct set of bins, it uses a *conservative threshold* that declares a bin *narrowband-occupied* if the power in that bin exceeds the noise floor by 3 dB in any sample, and *narrowband-free* otherwise. A sample is considered narrowband-occupied if any bin in that sample is narrowband-occupied.

SWIFT also uses its power measurements to compute four metrics that capture the most common responses to interference.

- *Inter-transmission time* captures the behavior of narrowband devices that react to interference by backing off (e.g. 802.11 or TCP backoff). It is computed by counting the number of consecutive narrowband-free samples.
- *Transmission duration* captures the behavior of devices that fall back to more robust, lower rate modulation schemes, thereby taking a longer amount of time for each transmission (e.g. autorate in 802.11). It is computed by counting the number of consecutive narrowband-occupied samples.
- *Average narrowband power* allows SWIFT to deal with multiple narrowband devices in the same band (e.g., two 802.11 devices). If SWIFT interferes with a nearby device causing it to backoff, but a more distant device fills in the freed bandwidth such that none of the other metrics changes, the average power will significantly decrease, allowing SWIFT to detect the change. This metric is computed by averaging the power in narrowband-occupied samples over a window.
- *Probability of transmission immediately after SWIFT* captures whether SWIFT triggers the carrier-sense reaction of narrowband. If SWIFT triggers narrowband carrier-sense, the narrowband device will not transmit immediately after a SWIFT packet, because it waits to ensure that the medium is free (In 802.11, this translates to the DIFS, followed by a random contention window). The metric is computed by looking at the power immediately after SWIFT finishes transmitting a packet, and setting a flag to 0 if the sample is narrowband-free, and 1 otherwise. The probability is computed as the average of these flags over a recent window.

SWIFT maintains sufficient statistics to compute the mean and variance of each metric. To achieve high confidence in the value of a particular metric, SWIFT needs to collect multiple measurements of that metric. Note that for the first three metrics, SWIFT gets one measurement every time it sees a narrowband transmission. The last metric is different, however, in that it can be measured independent of whether the narrowband device transmits or not. If the narrowband device has nothing to send though, the fact that no narrowband transmission is observed provides no information. Hence, SWIFT only includes samples of this metric when it senses a narrowband transmission within some maximum time after a SWIFT packet (1 ms in our implementation). Thus, the confidence of our estimates of all four metrics depends only on how many samples are obtained, and is independent of how sporadically the narrowband device transmits.

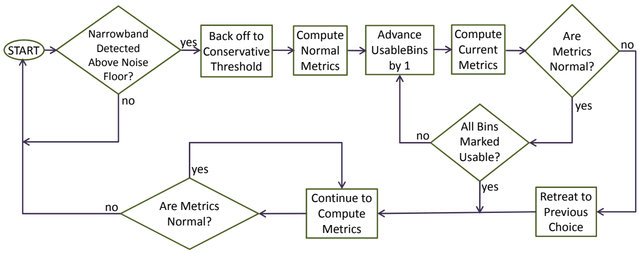


Figure 4: Control Flow for Adaptive Sensing Algorithm

5.1.3 Adaptive Sensing Algorithm

We define a bitvector `UsableBins`, which identifies the set of bins that SWIFT currently uses. The adaptive sensing algorithm starts with a conservative choice of `UsableBins` that does not interfere with the narrowband device, and iteratively tightens the setting of `UsableBins` to converge on the maximal set of usable bins that does not affect the narrowband device. Fig. 4 shows the control flow of our algorithm, which we describe in detail below.

Whenever SWIFT first detects narrowband power in a bin (using the conservative threshold), it immediately backs away from that bin, and updates `UsableBins` accordingly. This conservative choice of `UsableBins` allows SWIFT to be confident that observations made in this state represent normal narrowband behavior.

After gathering enough data at this normal setting, SWIFT begins the process of determining a choice of `UsableBins` that does not affect the narrowband device, but provides a maximal number of available bins. It starts by grouping contiguous sets of narrowband-occupied bins into a single *narrowband group*. Each narrowband group is then assigned a top and bottom bin which bound, for this narrowband group, the range of bins which must be left unused.

Next, SWIFT will try to grow `UsableBins` by using the top and bottom bins in each narrowband group and observing whether the narrowband device reacts. At each step, SWIFT alternates between reducing the top bin by one and increasing the bottom bin by one. For each choice of `UsableBins`, SWIFT waits to gather data measuring the effect of this new choice. It continuously monitors the incoming data by comparing the metrics with this bin choice to those observed under normal behavior with the conservative bin choice. If, at any point, SWIFT determines that it has impacted any of the metrics, it immediately moves back one step, and resets `UsableBins` to the previous decision. If, however, after gathering enough data, SWIFT determines that none of the metrics are impacted, it moves on to the next step, and tightens its choice further by one bin.

For each narrowband group, SWIFT independently continues this process until it either reaches a bin choice for which it notices the narrowband device reacting, in which case it retreats to the previous `UsableBins` setting, or it marks as usable all bins in this narrowband group and still notices no reaction. At this point, SWIFT continues to monitor the metrics and compare them to normal. If it notices a change at any point, SWIFT retreats to the conservative choice of `UsableBins`, recomputes normal metrics, and repeats the probing process, as shown in Fig. 4.

Note that this algorithm inherently deals with dynamics. For example, if the narrowband device moves closer or farther after SWIFT has finalized a bin choice, the average narrowband power metric will change from normal, and cause SWIFT to reinitiate the entire probing process. Furthermore, if all narrowband devices in a group depart, SWIFT will stop seeing any transmissions in the narrowband group, time out the entire group after a predefined interval, and reclaim these bins. Also, as articulated in §3, a narrowband device appearing in a new band currently occupied by SWIFT will always have the opportunity to transmit during SWIFT's idle intervals, and hence be quickly detected, allowing SWIFT to immediately back away and trigger the adaptive sensing algorithm for this new narrowband group.

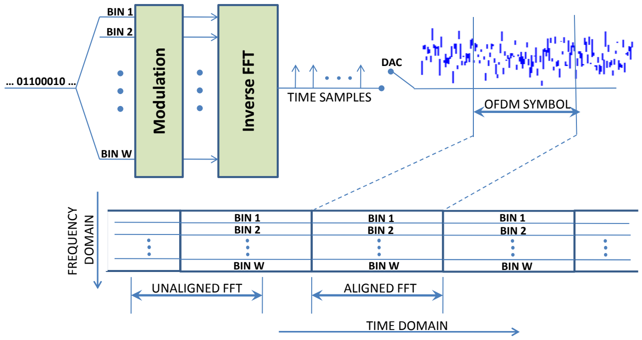


Figure 5: Conversion of bits into OFDM symbols: Values in individual frequency bins are combined in each time sample, and can be recovered only by computing appropriately aligned FFTs.

5.1.4 Measuring Statistically Significant Changes

When should SWIFT decide that changes in some metric are not due to statistical aberrations, but reflect a real change in the performance of the narrowband device?

SWIFT uses a statistical test called a *t-test*, typically used to decide whether a drug has had a statistically significant effect on the population studied [17]. A t-test takes the means, variances, and number of samples of the two compared sets: normal and current. It computes the following *t-value* where the $\bar{x}$'s and σ's represent the means and standard deviations, respectively, of the two sets, and n_{norm} and n_{curr} refer to the number of samples in the normal and current set:

$$t = \frac{\bar{x}_{norm} - \bar{x}_{cur}}{\sqrt{\frac{\sigma_{norm}}{n_{norm}} + \frac{\sigma_{cur}}{n_{cur}}}}.$$

To determine whether any difference between the means is statistically significant, the t-value must be combined with an alpha level, which represents the acceptable probability of being wrong. In our case, this value represents the probability that the t-test will tell us that SWIFT is interfering even if it is not. This is a parameter which effectively sets the aggressiveness of SWIFT. We use an alpha level of 0.05, typical for scientific and medical studies. The t-value combined with the alpha level and the total number of samples is then used in a table look-up to determine whether the t-test passes, i.e., whether SWIFT has had a statistically significant impact on narrowband.

5.2 Cognitive PHY

The cognitive PHY uses the output of adaptive sensing to provide a single high-throughput link over the set of usable bins.

On the transmitter, this means ensuring that no power is used in bins marked as narrowband-occupied by the adaptive sensing module. This is straightforward with OFDM since it naturally allows different power assignments for each frequency bin.

On the receiver side, the cognitive PHY has to ensure that the receiver can receive in non-contiguous bins even when narrowband devices are using the other bins. At first, it might seem that this can be done analogous to the transmitter by taking the FFT of the incoming signal, and just using values from the bins of interest. However, this is impractical. To understand why, consider the frequency-time diagram in Fig. 5 which illustrates how the N OFDM frequency bins are converted to N time samples that together represent an OFDM symbol. As can be seen, the correct frequency domain values can be retrieved from the time samples only when the FFTs are aligned correctly on OFDM symbol boundaries. But the receiver can align the FFT correctly on symbol boundaries only if it knows the starting sample of a packet in the first place!

Hence, we need to modify a few basic receiver algorithms to cope with non-contiguous bands.

(a) Receiver Packet Detection: In order to perform any processing on a packet, the receiver first needs to determine the start of the packet within a few time samples. Typically, this is done using the double sliding window approach [23], which uses energy ratios to determine the time sample where a burst of energy is received on the medium.

Since this operation happens in the time domain, it cannot distinguish between energy from narrowband devices and wideband transmitters, and can be spuriously triggered by narrowband transmissions. Recall that SWIFT concurrently transmits with narrowband devices by using separate frequencies. Hence, if the receiver is kept busy with false packet detections, it is very likely to miss desired wideband transmissions.[2]

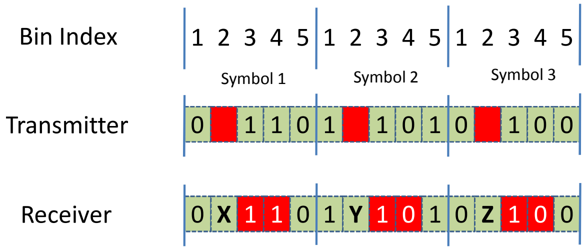
The solution is to actively filter the narrowband devices, allowing the receiver to perform packet detection on the clean signal consisting primarily of power from wideband transmitters. The choice of the bins to filter is driven by the adaptive sensing module. However, the receiver may not be able to use a filter per narrowband group since filters are resource-intensive in hardware. Hence, SWIFT is designed to use a small fixed number of bandstop filters, whose widths and center frequencies are dynamically configured. Note that since these filters are purely on the receiver side, by definition, they do not affect narrowband devices. A particular filter choice that is not perfectly aligned with the desired set of bins to be filtered only affects packet detection to the extent of the amount of narrowband energy that it lets in, or the amount of wideband transmitted energy it filters out. The filter computation problem is formulated as a dynamic program that eliminates as many narrowband bins as possible, while maximizing the amount of received wideband energy. The details of this optimization are omitted here for space, but described in [31].

(b) Receiver Packet Processing: Now that the start of the packet has been detected accurately, the receiver has the right alignment for the symbols and the rest of the packet processing can be done in the frequency domain over the actual bins used by the wideband system. Specifically, carrier frequency offset estimation, which is traditionally done in the time domain, is instead performed in the frequency domain after zeroing out the contributions of bins occupied by narrowband, as determined by adaptive sensing. This permits a more precise estimate than an application of the time domain estimation algorithms on the noisy filtered signal used for packet detection.

(c) Data reception: Recall that the transmitter, while assigning data to bins, zeros out all bins that are deemed unusable by adaptive sensing, and stripes data only across the remaining bins. Similarly, when the receiver collects the received data, it only utilizes bits from bins that are deemed unoccupied by narrowband devices. Again, we note that since data reception happens after the alignment provided by packet detection, it can work on the unfiltered signal and hence can precisely remove bins susceptible to narrowband interference.

5.3 Communication Over Uncertain Bands

Since each node in a SWIFT network independently decides the bands that it can use for transmission and reception, differences in proximity to narrowband devices and variations in time make it likely that a transmitter and receiver identify different bins as usable. For example, a wideband sender and receiver that are just a few meters apart may differ in their perspectives of narrowband-occupied bins by as much as 10-20 MHz as we show in §7.2.



Bin Index	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
	Symbol 1					Symbol 2					Symbol 3				
Transmitter	0	🟥	1	1	0	1	🟥	1	0	1	0	🟥	1	0	0
Receiver	0	X	1	1	0	1	Y	1	0	1	0	Z	1	0	0

Figure 6: Bin Disagreement Causes Communication Failure: If the transmitter sends in bins 1, 3, 4, and 5 while the receiver listens in 1, 2, and 5, then the receiver will decode noise in bin 2 as data, and miss data in bins 3 and 4. These insertions and deletions will cause a misalignment in the demodulated data stream, creating an error pattern than cannot be rectified by standard error-correcting codes.

This disagreement between a transmitter and its receiver can be a fatal obstacle to establishing an OFDM communication link. To understand why, recall that an OFDM transmitter stripes data across all usable OFDM bins. A receiver reconstructs the original data by extracting bits from the individual bins. Thus, as shown in Fig. 6, if the receiver expects data in a bin that the transmitter did not send in, it will result in insertion of bits into the data stream. Conversely, if the transmitter sends data in a bin that the receiver does not expect data in, it will manifest itself as deletions of bits from the data stream. Thus, disagreements about bins result in alignment and framing errors, and produce a wireless channel that has unknown insertions and deletions, which conventional error correcting codes cannot deal with.

We solve this problem using two mechanisms: (a) an infrequent synchronization phase when the communicating wideband pair has a drastic disagreement, say, when a wideband node boots up, or when many narrowband devices in different bands appear simultaneously, and (b) a low overhead handshake, which is used when nodes that have previously agreed experience a limited disagreement, say, because a single narrowband device was turned on or moved closer.

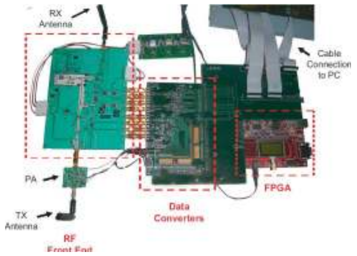
SWIFT nodes are equipped with a robust initial synchronization mechanism. Each SWIFT node divides the whole transmission band into chunks of 16 bins, checksums and codes the value of its `UsableBins`, and sends it simultaneously in all chunks. Assuming that the bandwidth of the wideband node is large enough, and has enough bins that are not interfered with narrowband, at least one of these chunks in this *sync packet* will be received correctly, allowing the nodes to establish connectivity. Note that the sync packet uses all OFDM bins, and hence does not suffer from an alignment problem.

Even after a SWIFT node pair is synchronized, they can still suffer from occasional disagreements, for example, when adaptive sensing changes the set of usable bins on a node. We leverage the existing agreement to transform the potential disagreements into bit errors, *i.e.*, we transform the hard problem of unknown insertions and deletions into the simpler problem of bit errors, a problem that all wireless links know how to deal with by adding practical error correcting codes.

To do so, SWIFT exploits the following key observation. If the transmitter stripes the data across the previously agreed bins, there will be no deletions or insertions. The problem, however, is that, by transmitting in the old bins, some of which may no longer be free, the transmitter might hinder a narrowband device. To address this problem, SWIFT stripes the data across the previously agreed

---

[2] Due to the hardware pipelining typical to receivers [23], they cannot receive packets while they are still working on the spuriously detected packet and have not rejected it.



Parameter	*Value*
Carrier Freq.	5.247 GHz
Data BW	100 MHz
Number of Bins	100 (×1 MHz)
Symbol Period	1.4 μs
Uncoded BER	10^{-3}
Bin Modulation	BPSK, 4-16-, 64-QAM
Max Link Len	10 m
Avg. Output Pwr	7.5 dBm

Figure 7: Wideband Radio Used in SWIFT

bins, but transmits only in the subset that is still usable. The receiver, which still expects to receive data across the old agreement, receives data in the intersection of the old and new bins correctly, but sees errors in the other bins. However, this can be easily fixed by using a simple error correcting code with sufficient redundancy to cover the expected extent of disagreement between old and new bins.

SWIFT uses a low-overhead handshake to quickly resolve disagreements. The data in the handshake is the new set of usable bins, and the striping technique is as described above. Once the handshake terminates, the nodes resume normal data exchange.

5.4 Network Issues

This section briefly describes how we compose multiple SWIFT links to build a network.

(a) The MAC: We use a carrier sense based MAC similar to 802.11 [22]. A node senses the medium and transmits if the medium is not busy. However, a direct application of the carrier sense technique of narrowband radios, which just checks for the total received power in the band to exceed a threshold, will unnecessarily reduce the transmission opportunities of SWIFT nodes since narrowband transmitters are always likely to be using some part of the band and hence preventing the wideband radio from transmitting. Instead, SWIFT's carrier sense focuses only on the bins declared usable by adaptive sensing. Specifically, when a node wants to send, it computes an FFT of the observed power, and proceeds with its transmission only if a large fraction of its usable bins are below the wideband carrier sense threshold.[3] Further, while wideband nodes can use an 802.11-like MAC, they need to wait for a relatively longer period to check that the medium is idle, *i.e.*, they should use a longer DIFS interval than typical values picked by narrowband devices. This ensures that a narrowband device that has just arrived into the environment can quickly access the medium and trigger adaptive sensing.

The SWIFT MAC randomly jitters the start of a probing epoch to ensure that different SWIFT nodes perform adaptive sensing independently. Further, a node uses control packets analogous to RTS/CTS to notify other SWIFT nodes of the start and end of a probing epoch in order to avoid simultaneous probing by multiple nodes. While this solution works for small wideband networks, extensions to larger networks may require more sophisticated mechanisms to leverage probing results across multiple SWIFT nodes.

(b) Transmitter Identification: The alert reader might have observed that a SWIFT receiver potentially needs to receive and decode packets from multiple transmitters; however, decoding a packet requires knowledge of the exact set of mutually agreed bins over which the data is striped, and this mutual agreement is likely to be different with different transmitters. Hence, the SWIFT receiver needs to identify the transmitter of a packet even before it can decode the packet. This is in contrast to current networks where a node decodes received packet headers to determine if they are intended for itself.

SWIFT adapts the technique of correlation with known pseudonoise sequences, typically used for packet detection, to develop a solution at the link layer. It is well known that pseudonoise sequences exhibit low correlation with each other while showing high correlation with themselves, thereby allowing identification of specific pseudonoise sequences purely by correlation [30]. Transmitter MAC addresses in SWIFT are pseudonoise sequences, and appear in a known and fixed symbol location in the received packet. When a receiver detects a packet, it correlates it against its neighboring nodes' MAC addresses to determine the transmitter, and hence the set of bins. This requires a receiver to maintain a table of neighbor MAC addresses; a receiver learns about a neighbor's MAC address during the initial sync packet where they exchange their mutually usable set of bins. Note that receiving the sync packet itself does not require prior bin agreement, as described in §5.3.

6 Implementing SWIFT

We have implemented SWIFT in a custom wideband radio transceiver platform developed by the WiGLAN research project [20]. The WiGLAN transceiver board, shown in Fig. 7, connects to the PC via the PCI bus, and acts like a regular network card. The transceiver [26] consists of three parts: 1) the RF front-end, which captures the analog signal, 2) the data converters, which convert between analog and digital, and 3) the digital baseband modem. All digital processing, such as packet acquisition, channel estimation etc., is done in baseband.

Our prototype has two components: the driver and the firmware. The former is implemented in software, and the latter in FPGA.

Driver: The driver presents a standard network interface to the kernel. In addition to this typical functionality, the driver offloads from the FPGA any computation that is too complex for hardware and is not on the critical path of an OFDM symbol. For example, the driver implements the metric computation and t-test (§5.1). Our current prototype implements two metrics: average narrowband power, and probability of transmission immediately after SWIFT.

Firmware: Several of SWIFT's major components that need to be on the critical path, such as narrowband power measurement (§5.1), the cognitive PHY (§5.2), the band consensus protocol (§5.3), and the MAC (§5.4), are implemented on the FPGA. We design SWIFT's algorithms in the Simulink environment, which has a hardware model for the Xilinx Virtex-4 SX35 FPGA that we use. The code is then compiled into an intermediate form using Xilinx tools [6]. We use Verilog to integrate this intermediate form with the PCI subsystem, and create the final hardware representation of our code.

7 Performance Evaluation

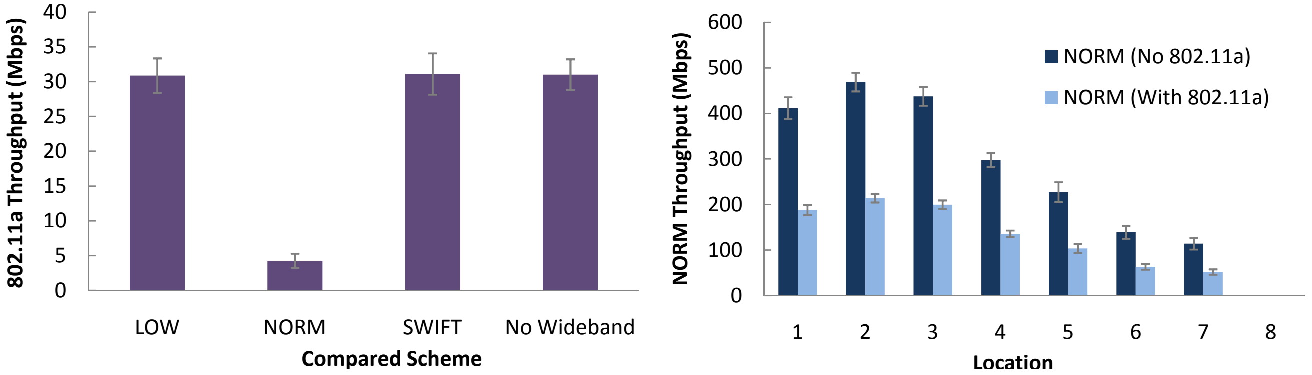
We evaluate SWIFT in a 12 node testbed consisting of four wideband nodes, and eight 802.11a nodes. Fig. 8 shows the experimental environment, which has high diversity due to the presence of walls, metal cabinets, desks, and various non-line-of-sight node locations. The exact choice of node locations for each experiment will be described along with the results for that experiment.

Wideband Devices. We use the WiGLAN wideband hardware described in §6, whose specifications are in Fig. 7. It has 100 OFDM data bins, numbered from -50 to +50, with bin 0 never being used. For all schemes, the wideband devices are evaluated while continuously sending 10 ms packets with a 1 ms gap between packets.

Narrowband Devices. These nodes run 802.11a in channel 52, corresponding to wideband bins 3 through 23. 802.11a nodes send UDP

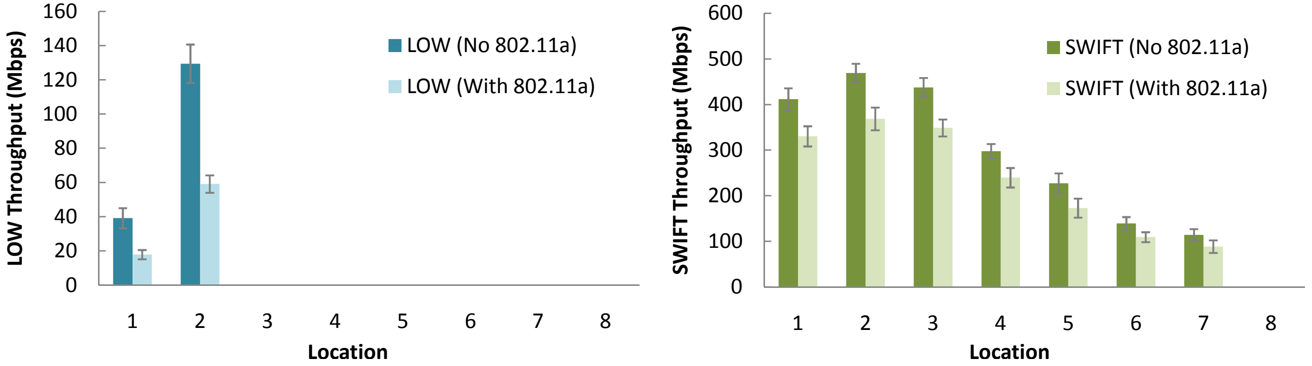
---

[3] Note that the objective of wideband carrier sense is not to correctly decode the received signal, but rather to measure received power, which does not require alignment.



(a) Throughput of 802.11a

(b) Throughput of Non-Adaptive Wideband (NORM)



(c) Throughput of Low-Power Wideband (LOW)

(d) Throughput of SWIFT

Figure 9: Approaches to Narrowband-friendliness: Presents the throughput-range tradeoff, and shows that SWIFT, illustrated in (d), is as friendly to 802.11a as LOW, while attaining dramatically higher throughput and operating range.

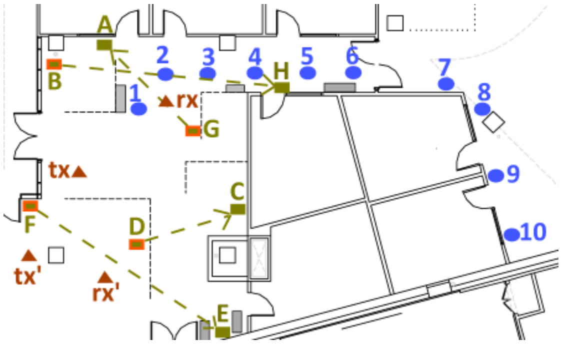


Figure 8: Testbed Map: Node Locations are Highlighted.

streams at the highest rate supported by the medium, except for experiment 7.5, in which they use TCP. The protocol, signal details, and occupied bands of 802.11a are, of course, unknown to SWIFT.

Compared schemes. We compare the different schemes by configuring our wideband hardware to run one of:

- **SWIFT**: This is the SWIFT protocol implemented as in §6.
- **Low-power wideband (LOW):** This is a baseline system that operates below the noise level to avoid interfering with narrowband devices. Specifically, it transmits signals with a power spectral density of -41 dBm/MHz, the FCC maximum for UWB devices [2].
- **Non-adaptive wideband (NORM):** This is a system that transmits across a wide band at the normal power of our hardware platform, but does not adapt to narrowband devices.

Note that both LOW and NORM will suffer drastic bit errors in bins used by 802.11a when it is turned on. For conservative comparison in this case, we therefore consider idealized versions of these systems that use the minimal amount of coding required to correct these errors.

7.1 Throughput and Range

This experiment explores if it is possible to be as narrowband-friendly as a transmitter operating below the noise level, while preserving the good throughput and range of a normal-powered wideband system.

Method. We place the wideband transmitter in location tx, and test its performance to the wideband receiver which is placed in each of locations 1 through 10. For each location, we measure the throughput of LOW, NORM, and SWIFT with and without interfering 802.11a traffic, and plot the results in Fig. 9.

Results. Fig. 9 demonstrates that, while both NORM and LOW are flawed, SWIFT can deliver on the fundamental goal of simultaneously achieving the high throughput and wide range of NORM, while being as narrowband friendly as LOW. In particular, we see that:

- **Throughput and range of LOW are limited:** Fig. 9(c) shows that LOW fails to get any throughput after location 2, and has $3.6 - 10.5\times$ lower throughput than SWIFT and NORM.
- **NORM is not narrowband friendly:** We can see from Fig. 9(a) that NORM significantly reduces 802.11a throughput.
- **SWIFT has high throughput and range:** From Figs. 9(b) and 9(d), we can see that in all locations, SWIFT achieves the same or greater throughput than NORM, with or without 802.11a.
- **SWIFT is narrowband friendly:** From Fig. 9(a), we can see that 802.11a throughput is unaffected by SWIFT.

We see from Figs. 9(b) and 9(d) that SWIFT surprisingly achieves higher throughput than NORM in the presence of 802.11a. This is because SWIFT intelligently avoids 802.11a occupied bins, while NORM uses these bins, suffers errors due to high narrowband power, and hence incurs additional overhead to correct errors in these bins.

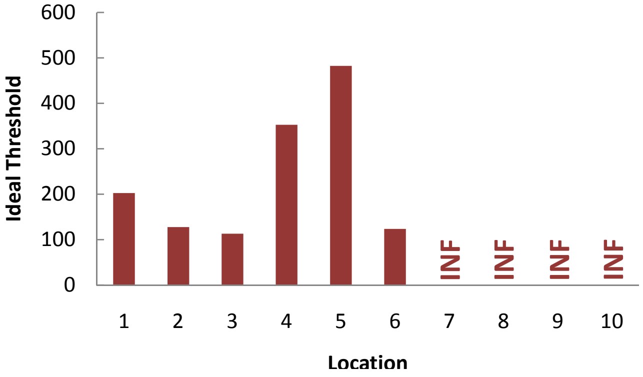


Figure 10: No Single Threshold Works Across Locations: This figure plots the ideal threshold that ensures safe narrowband operation while maximizing bins usable by wideband. 802.11a nodes at locations 7-10 are not affected by wideband, and hence the ideal threshold for these locations is infinity.

7.2 Power Threshold Sensing

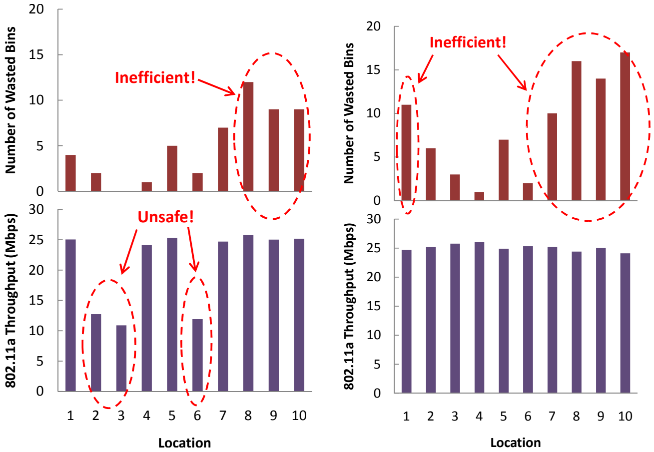
In §5.1, we discussed the intractability of a threshold based algorithm. Here, we present results validating that claim, first showing the difficulty of picking a threshold, and, second, showing that a single threshold cannot simultaneously be safe for narrowband, and efficient for wideband.

7.2.1 Difficulty in Using Thresholds

Method. This experiment uses one pair of SWIFT nodes at location tx and rx in Fig. 8, and one pair of 802.11a nodes which is moved among locations 1-10. At each location, we measure two quantities: (a) *Correct Bin Choice*: We disable adaptive sensing on SWIFT and manually try all possible usable bin settings until we find the maximal set of usable bins that does not affect 802.11a throughput.
(b) *Ideal Threshold*: This is defined for each location as the highest threshold that results in a bin choice which does not affect 802.11a in that location. This is the threshold that is most efficient for wideband, while still being safe for narrowband. We record the time average of the power SWIFT sees in each bin when 802.11a transmits, and calculate the ideal threshold as the minimum power across all bins that must be left unused to ensure safe 802.11a operation.
Results. Fig. 10 shows the difficulty in choosing a single threshold across locations: the ideal threshold varies by as much as 4.3× in our testbed; furthermore, the thresholds do not correlate with distance, because of the reflection and shadowing typical in an indoor environment.

7.2.2 No Single Threshold is Both Safe and Efficient

In this section, we illustrate how a particular choice of threshold forces a compromise between safe narrowband operation and efficient wideband performance across locations.
Method. We use the same placement of wideband nodes as in §7.2.1. We consider two thresholds based on our experiments in §7.2.1 above, setting the threshold to either the median, or the minimum of those in Fig. 10. We then determine the set of bins that would be marked as usable for each threshold setting and location. We disable adaptive sensing in SWIFT, and at each location, manually set it to use the set of bins resulting from the chosen threshold, and measure the 802.11a throughput.



(a) Median Threshold (b) Minimum Threshold

Figure 11: No Threshold is Safe and Efficient in All Locations

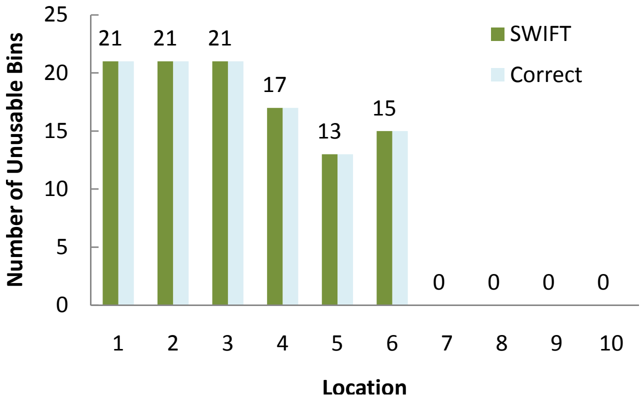


Figure 12: Adaptive Sensing is Robust: At each location, SWIFT finds the correct unusable bins, *i.e.* those that interfere with 802.11a.

Results. Fig. 11 compares the number of wasted bins, *i.e.*, bins that the threshold unnecessarily marks as unusable by wideband, at each location, against the corresponding 802.11a throughput, for both the median and minimum thresholds from Fig. 10. The median threshold leads to a dramatic reduction in 802.11a throughput in locations 2, 3, and 6, while simultaneously producing over 10 wasted wideband bins in each of locations 8, 9, and 10. Bins are wasted in these locations because the 802.11a nodes, being too far, are no longer affected by wideband transmissions, but this threshold still causes many bins to be marked as unusable. Note that a threshold-based design can be both unsafe and inefficient in the same location. In particular, with the median threshold it is unsafe in locations 2 and 6, but also wastes a few bins in those same locations. This is because a blip in power in any bin outside of those occupied by the narrowband device causes that bin to be wasted.

A lower choice of threshold would increase the likelihood of safe narrowband operation at the cost of increased inefficiency. For example, using the minimum threshold among all measured locations ensures safe 802.11a operation in all of these locations, but almost doubles the bandwidth wastage. In our example, in addition to wasting bins in locations 7, 8, 9, and 10 where 802.11a is out of range, it also wastes bins in location 1. This wastage is because 802.11a transmissions leak significant power into bins adjacent to those it uses. Additionally, this minimum threshold may be unsafe for locations outside the measured set, or for a different 802.11a transmitter.

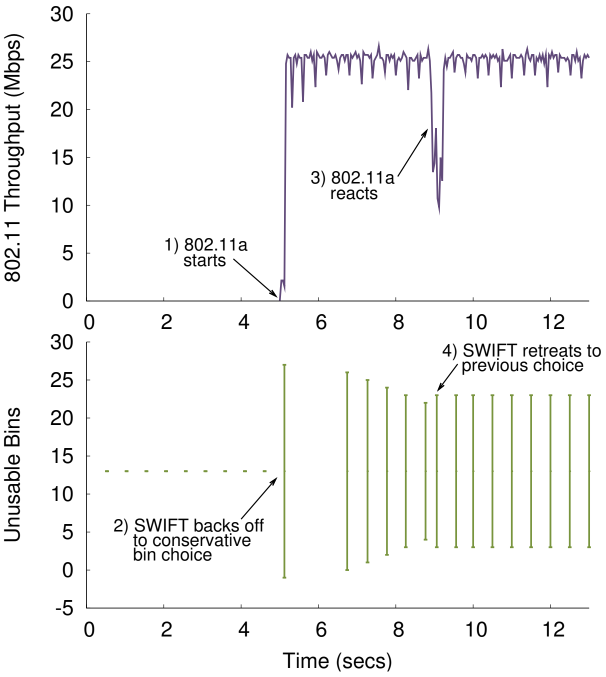


Figure 13: Responsiveness of Adaptive Sensing: The top graph shows that 802.11a throughput is not hindered for longer than 0.5 seconds by SWIFT. The bottom graph shows that, when 802.11a first appears, SWIFT backs off to a conservative bin choice within 120 ms, but quickly converges to a maximal set of safe bins.

7.3 Adaptive Sensing

In this section we show how the adaptive sensing algorithm allows SWIFT to use a maximal set of bins with almost no impact on 802.11a, and hence is both safe and efficient.

Method. The setup is similar to the previous experiment, except that the SWIFT nodes now have adaptive sensing turned on. We run one experiment at each location, by first starting the SWIFT node, and then starting the 802.11a transmission 5 seconds later. We record the `UsableBins` setting on which SWIFT settles, and compare it with the *correct* bin setting for each location as determined in §7.2.1.

Results. Fig. 12 shows that SWIFT finds the exact set of unusable bins, *i.e.*, bins that interfere with 802.11a, at all locations. Note further that SWIFT detects when 802.11a goes out of range, as in locations 7-10, and can reclaim all occupied bins.

Fig. 13 shows the typical dynamics of adaptive sensing, using results from an experiment with 802.11a at location 3. SWIFT conservatively backs away from bins used by 802.11a within 120 ms of 802.11a commencing transmission. Additionally, within 4 seconds, it finds the ideal bin selection and then sticks with this selection. Over 60% of this time is a result of the communication overhead from our prototype PCI driver, and can be mostly eliminated with an optimized implementation.

Specifically, the bottom graph shows the SWIFT bin selections over time. SWIFT starts out using all bins, (1) until it first detects the 802.11a transmissions. (2) At this point, SWIFT immediately backs off using a conservative threshold, and avoids bins -2 through 28. As it gathers more data, and determines that 802.11a is unaffected, SWIFT decreases its set of unused bins gradually, till it begins avoiding only bins 4 through 22. (3) At this point, we see from the top graph that the throughput of 802.11a is affected for the first time. (4) SWIFT immediately relaxes its bin selection to avoid bins 3 through 23, and this returns the throughput of 802.11a to normal. As a result, SWIFT stabilizes at a state that avoids bins 3 through 23, which is the tightest bin selection that does not affect 802.11a.

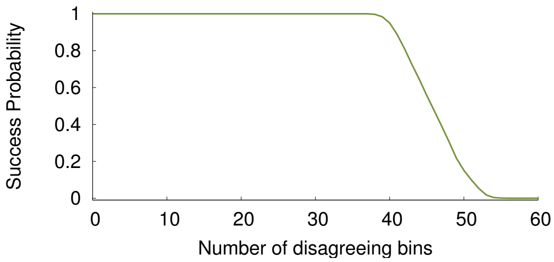


Figure 14: Robustness to disagreement: The figure shows the probability of a transmission succeeding as a function of the number of disagreeing bins. It shows that SWIFT is robust to as much as 40% disagreement between the set of transmitter and receiver bins.

7.4 Dealing with Bin Disagreement

We evaluate the impact of disagreement between communicating pairs on SWIFT's band consensus protocol.

Method. We place the wideband transmitter and receiver within a few feet of each other so that they can communicate with each other with very low probability of channel bit errors. We do this to ensure that almost all bit errors are likely to be introduced purely due to disagreements. We initialize the transmitter and receiver to agree to use the entire wide band, consisting of 100 bins.

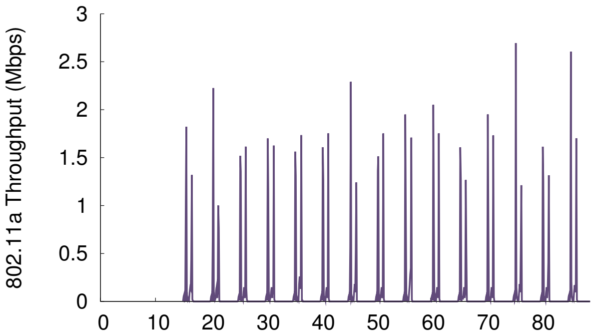
We then configure the adaptive sensing module to update the transmitter with a new set of usable bins with a sequence of K consecutive bins marked as narrowband-occupied, to simulate the appearance of a narrowband transmitter with a band of size K. Since the transmitter cannot use these bins whereas the receiver continues to expect data in them, the size of the disagreement between the nodes is K. We send a random coded sequence from transmitter to receiver using this disagreeing set of bins, check whether it is received correctly, and repeat this operation with a large number of random coded sequences for increasing values of K. We declare a transmission to have succeeded if it is decoded correctly, and compute the probability of a successful transmission for a disagreement of size K.

Results. Fig. 14 shows that SWIFT's band consensus works robustly for a large range of disagreements. When K is small, the consensus scheme sees a very small number of errors which can be easily corrected. As K grows, the receiver sees a burst of errors in the disagreeing bins, but the number of errors in any single code word is limited because transmitted data bits are interleaved across the frequency bins. This allows successful transmissions even when the fraction of disagreement is as large as 37% (37 of the 100 total bins). Such a large amount of disagreement is extremely unlikely, and hence SWIFT's low overhead handshake mechanism can almost always achieve band consensus. It is only when the extent of disagreement becomes large (56 bins in our case) that SWIFT nodes will need to reestablish connectivity using a sync packet.

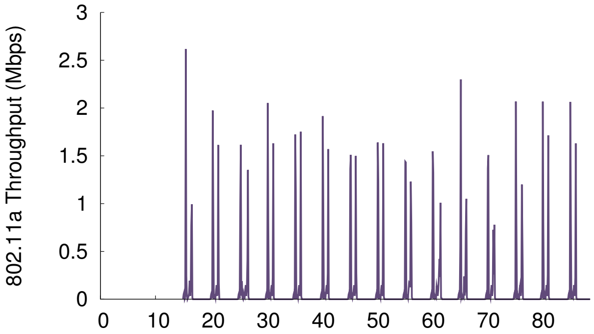
7.5 Intermittent Narrowband TCP Web Downloads

This experiment evaluates SWIFT's ability to adapt correctly to intermittent and bursty traffic patterns.

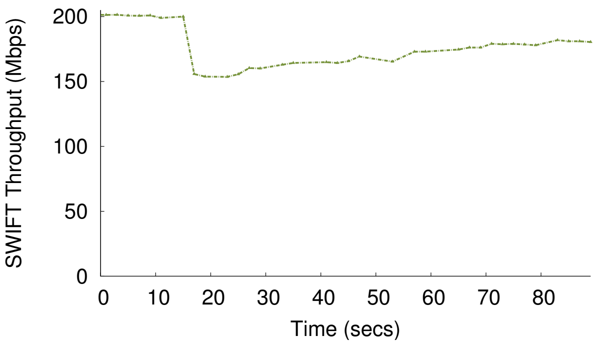
Method. We model a typical home scenario, using an 802.11a node that accesses the Internet by connecting to a Linksys wireless router. We first start the SWIFT node, and at time $t = 15$ seconds, the 802.11a node begins periodic web downloads. For this experiment,



(a) 802.11a Throughput without SWIFT



(b) 802.11a Throughput with SWIFT



(c) SWIFT Throughput with TCP

Figure 15: SWIFT reaction to TCP web downloads: (a) and (b) show that, even in the face of intermittent 802.11a traffic, SWIFT avoids affecting 802.11a transmissions, while (c) shows that it does this while still achieving 90% of its original throughput.

we download the home page from `www.apple.com` every 3 seconds. We average the throughputs of the TCP downloads and SWIFT over 100ms intervals, and plot them as a function of time.

Results. Fig. 15 shows that SWIFT adapts to intermittent and bursty web traffic, without causing any performance impact on the narrowband user. Notice that the narrowband traffic is indeed intermittent, and that the TCP downloads are too short for narrowband to achieve a peak throughput higher than 2-3 Mbps, despite the fact that the auto-rate algorithm is sustaining 48 or 54 Mbps in this case.

We see that SWIFT throughput drops as soon as the user begins her web download. This is because SWIFT falls back to a conservative set of bins. SWIFT throughput then gradually increases as it tightens its set of bins. However, this process is slower than the example in Fig. 13 because SWIFT only uses measurements in the vicinity of a narrowband transmission, as described in §5.1. It therefore needs to wait for a longer time to acquire enough data points for each bin choice. SWIFT converges on the right set of bins, and its throughput stabilizes around $t = 75$ seconds. This throughput is lower than the throughput that SWIFT achieved prior to the web downloads because SWIFT is now avoiding bands that could affect 802.11a performance. Throughout this process, SWIFT remains safe to 802.11a and does not cause any noticeable impact on the TCP throughput.[4]

7.6 Network Results

Here, we show that SWIFT performs well even in a chaotic environment with multiple 802.11a devices, and multiple SWIFT nodes.

Method. In this experiment we use four wideband nodes and eight 802.11a nodes, creating six pairs of communicating nodes. We place the four 802.11a pairs at locations A-H, and the two wideband pairs at the locations labeled tx/rx and tx'/rx' in Fig. 8. We then measure the throughputs when running the network without any wideband transmitters, with the wideband transmitters running NORM, and with the wideband transmitters running SWIFT.

Results. Fig. 16(c) shows that, when NORM transmits simultaneously with 802.11a, it significantly reduces 802.11a throughput. While the throughput reduction of 802.11a pairs at different locations is different, all pairs are impacted, with an overall average loss in throughput of around 50%.

Figs. 16(a) and (b) show the throughput of the four 802.11a pairs, with and without SWIFT. In this case, both pairs of SWIFT nodes move away from the bins occupied by the 802.11a nodes, allowing all 802.11a pairs to have essentially the same performance as in the absence of SWIFT. Additionally, Fig. 16(c) shows that by utilizing all bins not occupied by 802.11a, the SWIFT nodes are each still able to get reasonable throughputs of 30-100 Mbps in the face of 802.11a.

This result shows that SWIFT can deliver an operational wideband network, while ensuring that it does not affect multiple competing narrowband nodes.

8 Conclusion

This paper addresses the problem of coexistence between emerging wideband networks and narrowband devices with which they share the unlicensed bands. We show that overly conservative designs that avoid interference by running below the noise floor needlessly sacrifice the throughput and the range of the wideband radios. In contrast, a design based on cognitive aggregation, which adapts its frequency bands and weaves together multiple non-contiguous bands into one wireless link, can be as narrowband-friendly as the conservative approaches, while achieving a significant increase in operating range and throughput.

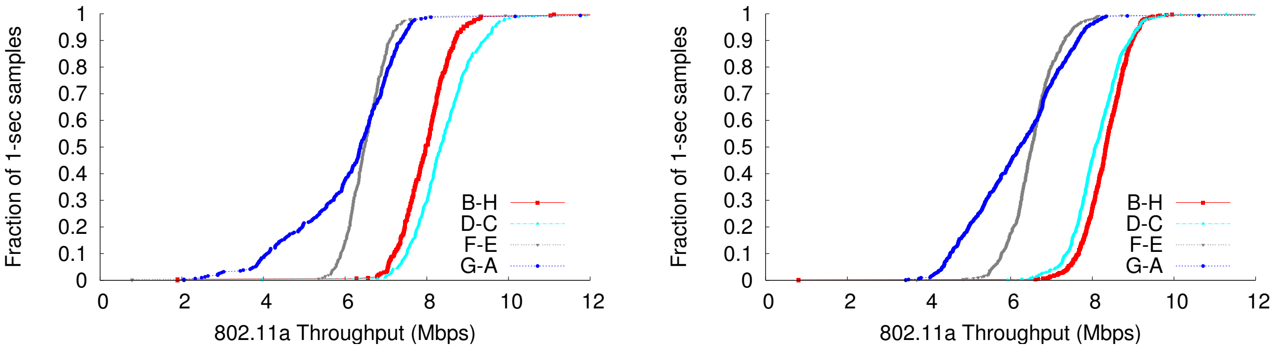
Our results can be extended in multiple directions:

(a) Non-reactive narrowband devices: This paper addresses narrowband technologies that react to interference in their band. Of course, not all devices react to interference. We envision that SWIFT can be extended to deal with such devices in one of two ways: either by being configured to avoid known non-reactive bands if they are present, or by having adaptive sensing recognize a device as non-reactive if all narrowband bins can be reclaimed without any identifiable reaction. In this case, SWIFT can fall back to a conservative bin setting that avoids all bins with non-reactive narrowband power.

(b) Coexistence of multiple wideband protocols: SWIFT selectively avoids frequency bands used by narrowband devices, and

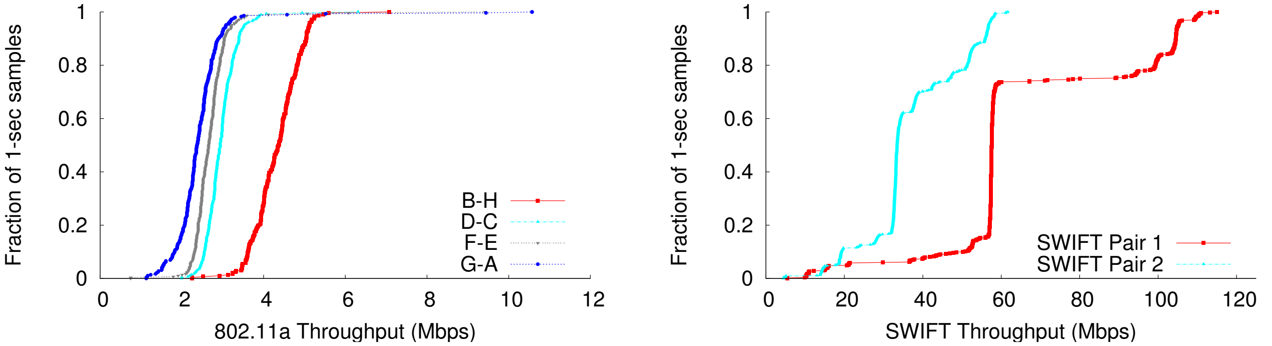
---

[4] The differences in TCP throughput with and without SWIFT are caused by varying queue lengths in the wired Internet. In particular, note that the variations in downloads between the two graphs are no greater than the variations within any one graph.



(a) 802.11a Throughput alone

(b) 802.11a Throughput with SWIFT



(c) 802.11a Throughput with NORM

(d) SWIFT concurrent with 802.11a in (b)

Figure 16: Throughputs in a Network: (a) and (b) show the throughputs of the four 802.11a pairs, with and without SWIFT. SWIFT has no impact on 802.11a, while, still getting good throughput as seen in (d). In contrast, (c) shows that non-adaptive wideband transmitters reduce 802.11a throughput by around 50%.

shares the spectrum with other cooperating wideband devices using the SWIFT protocol. However, the future may bring a variety of wideband protocols. These systems need to find a way to share spectrum among different wideband technologies even when they do not use the same protocol.

(c) Dynamic Range: Like other techniques that allow a node to receive multiple concurrent signals [33], SWIFT's nodes deal with a wide range of signal powers and hence their performance improves with a wider dynamic range of the system.

Acknowledgments: We thank Manish Bharadwaj for several insightful discussions on the design of SWIFT, and Szymon Chachulski, Micah Brodsky, and Rob Beverly for their helpful comments on the paper. This work is funded by DARPA ITMANET and Quanta.

References

[1] Cutting the Cord to Flat-screen TVs. CNN Jan 2008.

[2] FCC Slides for UWB Spectral Limits. `http://sss-mag.com/uwbslides.html`.

[3] IEEE 802.22 WG. `www.ieee802.org/22/`.

[4] Impact of devices using UWB technology on radiocommunication services. ITU R SM.1757.

[5] UWB - Intel Standards. `http://intel.com/standards/case/case_uwb.htm`.

[6] Xilinx. `http://xilinx.com/products/design_resources/design_tool/`.

[7] Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY), 1999.

[8] Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY), 2003.

[9] Enabling high-speed wireless personal area networks, 2005. Intel White Paper.

[10] I. F. Akyildiz, W. Y. Lee, M. Vuran, and S. Mohanty. NeXt Generation Dynamic Spectrum Access Cognitive Radio Wireless Networks: A survey.

[11] J. G. Andrews, A. Ghosh, and R. Muhamed. *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*. PrenticeHall, 2007.

[12] P. Bahl, R. Chandra, P. A. Chou, J. I. Ferrell, T. Moscibroda, S. Narlanka, and Y. Wu. KNOWS: Kognitiv Networking Over White Spaces. In *IEEE DySPAN 2007*.

[13] H. Bölcskei. *Principles of MIMO-OFDM wireless systems*. 2004.

[14] V. Brik, E. Rozner, S. Banerjee, and P. Bahl. DSAP: A Protocol for Coordinated Spectrum Access. In *IEEE DySPAN 2005*.

[15] M. M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access. 2005.

[16] D. Cabric, S. M. Mishra, D. Willkomm, R. Brodersen, and A. Wolisz. A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum. In *14th IST Wireless Comms. Summit*.

[17] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury, 2nd edition, 2002.

[18] M. S.-W. Chen and R. Brodersen. A Subsampling UWB Impulse Radio Architecture Utilizing Analytic Signaling. *IEICE Transactions on Electronics, Vol, E88-C*, 2005.

[19] S. S. Company. Spectrum occupancy measurement, 2007. `http://www.sharedspectrum.com/measurements/`.

[20] F. Edalat. *Real-time Sub-carrier Adaptive Modulation and Coding in Wideband OFDM Wireless Systems*. PhD thesis, Massachusetts Institute of Technology, 2008.

[21] S. C. Ergen. ZigBee/IEEE 802.15.4 summary, 2004.

[22] M. S. Gast. *802.11 Wireless Networks*. O'Reilly, 2nd edition, 2005.

[23] J. Heiskala and J. Terry. *OFDM Wireless LANs: A Theoretical and Practical Guide*. Sams Publishing, 2001.

[24] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM '88*, Stanford, CA, 1988.

[25] L. Ma, X. Han, and C.-C. Shen. Dynamic Open Spectrum Sharing MAC Protocol for Wireless Ad Hoc Networks. In *IEEE DySPAN 2005*.

[26] N. Matalon. An Implementation of a 5.25 GHz Transceiver for High Data Rate Wireless Applications. MS thesis, MIT, EECS, July 2005.

[27] M. McHenry. Frequency Agile Spectrum Access Technologies. Presentation to 2003 FCC Workshop on Cognitive Radios.

[28] S. Mishra, S. ten Brink, R. Madadevappa, and R. Brodersen. Detect and Avoid: An Ultra-Wideband/WiMax Coexistence Mechanism. *IEEE Communications Magazine*, June 2007.

[29] M. Mittelbacht, C. Mullert, D. Fergert, and A. Fingert. Study of Coexistence Between UWB and Narrowband Cellular Systems. In *UWB Systems*, 2004.

[30] J. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill, 5th edition, 2007.

[31] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat. SWIFT: A Narrowband-Friendly Cognitive Wideband Network. Technical Report MIT-CSAIL-TR-2008, MIT, 2008.

[32] R. Tandra and A. Sahai. SNR walls for signal detection. In *IEEE Journal on Special Topics in Signal Processing*, Feb. 2008.

[33] D. Tse and P. Vishwanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[34] C. Wilmot. Intel demonstrates fast new Ultrawideband WPAN at IDF Taiwan, 2006.

[35] J. Zhao, H. Zheng, and G. Yang. Distributed Coordination in Dynamic Spectrum Allocation Networks. In *IEEE DySPAN 2005*.

[36] Q. Zhao, L. Tong, and A. Swami. Decentralized cognitive MAC for dynamic spectrum access. In *IEEE DySPAN 2005*.

[37] ZigBee Alliance. *ZigBee and Wireless Radio Frequency Coexistence*, June 2007. White Paper.