Fine-grained Spectrum Adaptation in WiFi Networks

Sangki Yun, Daehyeok Kim, Lili Qiu
The University of Texas at Austin
{sangki,dkim,lili}@cs.utexas.edu

Abstract – Explosive growth of WiFi traffic calls for new technologies to dramatically improve spectrum efficiency. In this paper, we propose an approach to adapt the spectrum on a per-frame basis. It consists of three major components: (i) a fine-grained spectrum access design that allows a sender and receiver to change their transmission and reception spectrum on demand, (ii) fast and accurate spectrum detection that allows a receiver to determine which spectrum is used by its sender on a per-frame basis by exploiting the IEEE 802.11 preamble structure, and (iii) an efficient spectrum allocation algorithm that determines which spectrum to use for each transmission by taking into account frequency diversity and interference. It can further be adapted to perform a joint assignment of spectrum, schedule, and access point (AP) for each frame. Using a SORA implementation and trace-driven simulation, we demonstrate the feasibility of per-frame spectrum adaptation and its significant benefit over existing channel assignment approaches. To our knowledge, this is the first per-frame spectrum adaptation prototype for WiFi networks.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Experimentation, Performance

Keywords

IEEE 802.11, Dynamic Spectrum Access, Spectrum Allocation, Physical Layer

1. INTRODUCTION

Motivation: As WiFi traffic has grown explosively, new technologies have to be developed to dramatically improve spectrum efficiency and keep up with the users' ever increasing traffic demands. In response to this need, a recent trend in wireless standards is to adopt wider spectrum to increase data rates. For example, IEEE 802.11n [20] supports up to 40 MHz channels and IEEE 802.11ac [24] further increases the channel width up to 160
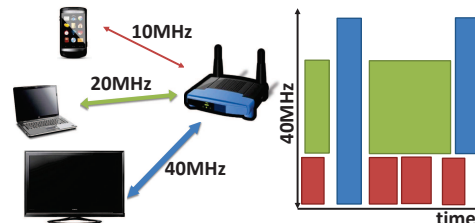
Figure 1: Fine-grained spectrum adaptation (FSA) example scenario.

MHz. However, simply using larger channel width is not always desirable. For instance, IEEE 802.11n has a known problem that a node using a 40MHz channel can be easily starved by nodes using 20MHz channels [5, 35]. Moreover, a wider channel reduces the communication range due to less concentrated energy [3] and also increases energy consumption due to a higher sampling rate. In contrast, WIFI-NC [5] advocates using narrowband channels to reduce MAC overhead as well as improve energy efficiency at the cost of more guard bands or reduced SNR.

Instead of fixing to a narrowband or a wideband, we believe the right approach is to determine spectrum based on the application need and current channel condition. By doing this, we can achieve the best of both worlds. This concept is illustrated in Figure 1, where an AP communicates with a mobile device on a 10MHz channel to conserve power and achieve a higher SNR over the same distance, communicates with a laptop on a standard 20MHz channel to balance power and bandwidth requirements, and communicates with HDTV on a wider channel to achieve high capacity. Note that the AP here may be either a sender or receiver. When it acts as a sender, it should be able to quickly switch the channel on a per-frame basis depending on which receiver the current frame is destined to; when it acts as a receiver, it should be able to not only quickly switch the spectrum but also detect which spectrum the incoming transmission uses on a per-frame basis so that it can switch to the right spectrum to receive the incoming signals. The need for dynamic spectrum adaptation further increases when the best channel changes quickly (*e.g.*, due to fast fading or dynamic traffic). If the receivers can promptly feed back the channel quality, which is made feasible with low overhead thanks to recent advances in feedback technologies, such as feedback compression [15], full duplex radios [6, 12], and embedding feedback in the data traffic [7], the transmitters can immediately adapt the spectrum to achieve high performance.

Challenges: In order to support the above functionalities, we need the following three fundamental capabilities:

- A sender or receiver should promptly switch to the desired spectrum. Tuning the spectrum parameter (*e.g.*, center frequency and channel width) in hardware incurs too long delay.

- A receiver should accurately detect the spectrum used by the transmission within a preamble time. There are many possible combinations of channel width and center frequency, which makes accurate and fast detection challenging.

- Given the capability of fine-grained spectrum access, it is important to determine which spectrum to use for each transmission to optimize performance and avoid interference.

Our approach: To address these challenges, we develop Fine-grained Spectrum Adaptation (FSA) that allows (i) a transmitter and a receiver to dynamically change the center frequency and channel width on a per-frame basis and (ii) a receiver to quickly identify the spectrum used by the sender based on the IEEE 802.11 preamble signature. In particular, a sender freely selects its channel width and center frequency without coordination with its receiver. It uses flexible baseband design to virtually change the spectrum, and optionally transmit multiple narrowband signals using one wideband radio. The receiver detects the transmission within the preamble detection phase using a novel spectrum detection method based on the special characteristics of the preamble in the frequency domain. Then it performs spectrum shaping, sampling clock and filter adjustment, and compensation for the center frequency difference of data symbols while fixing the hardware parameters. If multiple signals are transmitted, the receiver can decode all the signals from different subbands.

Built on the capability of per-frame spectrum adaptation, we then develop an algorithm to select the spectrum to use for each transmission according to the current channel condition. Different from OFDMA scheduling, which targets synchronous transmissions in cellular networks, our spectrum allocation works for unsynchronous transmissions (*i.e.*, transmissions from the same or different nodes are not synchronized) as common in WiFi networks. Our algorithm takes into account both frequency diversity and interference, and can support *joint* allocation of spectrum, schedule, and AP. Our algorithm is flexible: (i) an AP can use different spectrum to communicate with different clients or with the same client at different time depending on the current channel condition, and (ii) a frame can be allocated to non-contiguous spectrum or even to different APs if it improves performance.

Our main contributions can be summarized as follow:

- A baseband design that allows a sender and receiver to quickly change their frequency bands. It is fully compatible with 802.11a PHY: an FSA enabled device can communicate with a legacy 802.11a device that does not know FSA (Section 3.1).

- A simple yet effective algorithm that lets a receiver accurately determine which spectrum is used by the sender and how it is used by exploiting the IEEE 802.11 preamble structure (Section 3.2).

- An algorithm that selects the appropriate spectrum to use for each transmission by taking into account both frequency diversity and interference (Section 4).

- A SORA implementation that demonstrates the feasibility of spectrum adaptation on a per-frame basis and its performance benefit, and trace-driven simulations that show its effectiveness in more general scenarios. To our knowledge, this is the first WiFi compatible per-frame spectrum adaptation prototype (Section 5 and 6).

2. RELATED WORK

Our work is closely related to the following three broad areas: (i) spectrum virtualization, (ii) dynamic spectrum access, and (iii) spectrum allocation.

Spectrum virtualization: The signal bandwidth has long been considered to be determined by RF front-end, which takes non-negligible delay to change. In commercial hardware, changing the channel width takes over 1 ms; and even in advanced RF hardware, it takes more than 100 μs [27]. Recently, spectrum virtualization techniques change the spectrum in baseband level while fixing the RF bandwidth [10, 32]. Using signal shaping techniques such as sampling rate conversion and frequency shifting, it changes spectrum with little overhead. This allows a sender to use different spectrum on a per-frame basis. However, [10, 32] do not consider how the sender and receiver agree which spectrum to use. Rodin [2] develops a new preamble for a receiver to identify the spectrum used by the sender. Different from Rodin, our approach is compatible with WiFi by leveraging the existing preamble in IEEE 802.11.

Dynamic spectrum access: Ranveer et. al. [3] show that dynamically changing channel width gives better throughput than using a static wide channel, and proposes SampleWidth to dynamically adjust the channel width. As it relies on RF for channel switching, the switching delay is significant. To reduce overhead, it adapts the channel width only infrequently. Moreover, an AP has to use the same channel width for all clients. Our FSA overcomes these issues by using faster and more flexible spectrum adaptation.

WiFi-NC [5] proposes using narrow channels to improve spectrum efficiency. This is based on the following insights: (i) the relative overhead of PHY/MAC is smaller in narrower channels, (ii) it is challenging to efficiently support heterogeneous channel widths, and (iii) the spectrum may be fragmented and only leave narrow channels available. However, narrow channels tend to incur a higher overhead due to more guard bands. While WiFi-NC uses sharp elliptic filters to achieve narrow guard bands, it incurs preamble dilation and requires a larger CP.

FICA [31] also advocates narrower sub-channels. Different from WiFi-NC, the subchannels in FICA are dependent and transmissions across different sub-channels should be synchronized within a few microseconds. Therefore, as pointed out in [5], FICA is essentially a wide single-channel system with multiple inter-dependent narrow sub-channels.

SWIFT [26] and JELLO [34] dynamically allocate spectrum depending on the narrowband interference or the traffic demand, but they require coordination for the spectrum agreement. ASN [35] allows adaptive subcarrier selection and detection using the preamble. But it assumes all nodes use the same OFDM structure and is only applicable when they use the same RF bandwidth. Also, its detection method is similar to STD, which significantly underperforms FSA as shown in Section 5.

The emerging IEEE 802.11ac standard supports up to 160MHz wide channel communication and provides dynamic bandwidth management mode that allows 20MHz subchannel communication [24]. It uses RTS and CTS exchange for spectrum agreement and incurs significant overhead.

Spectrum allocation: WhiteFi [1] provides WiFi like connectivity in TV spectrum. Its detection algorithm exploits different ACK time duration under different channel width. It requires storing the signal and defers decoding till the end of ACK reception, which increases decoding delay and complicates system design. In comparison, FSA detects the bandwidth using the preamble and the receiver can immediately start decoding as soon as it receives data signals. Moreover, the detection algorithm in WhiteFi works only when the

center frequency of the sender and the receiver are aligned, which is not required by FSA.

FLUID [27] dynamically allocates spectrum. To address significant channel switching overhead, it uses two WiFi interfaces to hide the switching delay and also switches channel less frequently. Our work is inspired by FLUID and benefits from their proposed interference model, but differs from it in the following ways: (i) our spectrum adaptation is done on a per-frame basis (a few microseconds in FSA vs. above 4 ms in FLUID), (ii) our flexible baseband allows an AP to transmit multiple frames simultaneously on different spectrum, whereas FLUID allows an AP to transmit only one frame at a time, which limits spectrum efficiency, and (iii) we support allocating a frame to non-contiguous spectrum and perform joint AP assignment and spectrum adaptation.

3. FINE-GRAINED SPECTRUM ACCESS

We develop fine-grained spectrum access. It consists of a baseband design for a sender and receiver to quickly change the spectrum and spectrum detection scheme for a receiver to identify the spectrum used by the sender. Once the receiver detects the spectrum, the standard decoding is used to decode signals on the detected spectrum.

3.1 Baseband Design

We first introduce the baseband system design of FSA. As Analog to Digital Conversion (ADC), Digital to Analog Conversion (DAC), baseband to passband conversion, and passband to baseband conversion are performed in inflexible Radio Frequency (RF) front-end hardware, we describe how to implement FSA using flexible baseband signal processor. Our design goal is to support efficient spectrum change with negligible overhead. In this paper, since the maximum bandwidth of SORA is limited to 20MHz, we assume the physical bandwidth of the radio is 20MHz and confine possible channel widths to 5, 10, and 20MHz, but our system design is general enough to apply to wider band systems.

One easy way to communicate through narrow channel is to use a small number of subcarriers for data transmission in OFDM systems. Examples of such systems include FICA [31] and OFDMA. This approach is inflexible because it works only when transmitters in different spectrum are tightly synchronized with very small misalignment [5].

Another option is to use narrower subcarrier width. IEEE 802.11a standard uses this approach to support 5MHz and 10MHz channels in addition to the default 20MHz channels. Most commercial WiFi devices support it. This is also what we use in our system. By reducing the subcarrier width, most of the timing related values increase inversely proportional to the reduction ratio, while the basic PHY architecture remains the same. For example, an OFDM symbol duration is $4\mu s$ in 20MHz channel, and increases to $16\mu s$ in 5MHz. While the number of subcarriers remains constant (*i.e.*, 64), the bandwidth of each subcarrier is adjusted proportional to the channel width. For example, the subcarrier widths in 20MHz and 5MHz channels are 312.5KHz and 78.1KHz, respectively. The benefit of this approach is that nodes in different spectrum can independently communicate without the need to synchronize. However, most commercial devices incur non-negligible delay in changing the channel width, so it is too costly to change on a per-frame basis. Instead we use upsampling and downsampling to adjust subcarrier width without increasing delay.

Transmitter: Figure 2 (a) shows the baseband system design to send multiple narrowband signals using one wideband RF hardware. Based on the available channel and the application requirement, the sender decides which spectrum to use. If the channel
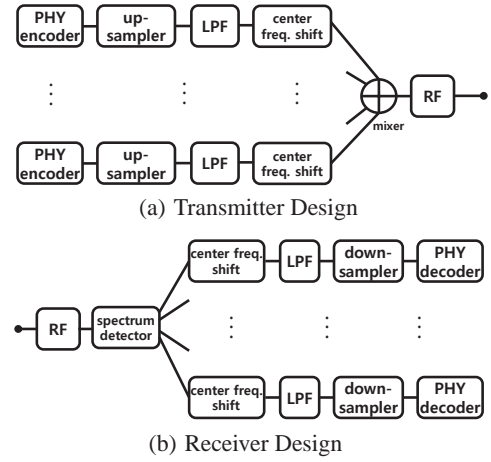

(a) Transmitter Design

(b) Receiver Design

Figure 2: Baseband System Design.

width is determined to be 5MHz or 10MHz, instead of changing the hardware sampling clock in RF-frontend, it performs upsampling in the baseband processor. Assuming that the signal rate is fixed, upsampling process adds interpolated samples for each signal. This increases the signal duration, which in turn decreases the spectrum width of the signal [23]. More specifically, upsampling by a factor of N increases the sampling rate to N times by adding N - 1 interpolated samples for every signal and decreases the bandwidth to the ratio of $\frac{1}{N}$. Using the baseband signals generated by 802.11a PHY, we generate 10MHz and 5MHz bandwidth signals through upsampling by factors of 2 and 4, respectively. As upsampling process generates image of the signal outside the band, Low Pass Filtering (LPF) is required to avoid generating adjacent channel interference [23].

In addition to changing the channel width, if the transmitting spectrum and RF hardware have different center frequency, we need to shift the center frequency as follows. Suppose that the center frequency of the RF hardware is tuned to f_c, but the sender wants to use the center frequency $f_c + \Delta$. This difference can be compensated in the baseband level as follow. Let $s[n]$ be the baseband signal that the sender transmits, and Δ be the difference between the desired and the hardware center frequency. We generate frequency shifted signal $s_{fs}[n]$ by multiplying $e^{j2\pi\Delta n}$ to each sample $s[n]$ (*i.e.*, $s_{fs}[n] = s[n]e^{j2\pi\Delta n}$). In RF hardware, this signal is converted into passband signal $x[n] = s_{fs}[n]e^{j2\pi f_c n} = s[n]e^{j2\pi(f_c+\Delta)n}$, which is the desired signal with center frequency $f_c + \Delta$.

If the sender transmits multiple signals through more than one subband, it unsamples all signals to 40MHz since the widest possible signal in our implementation is 20MHz and should be upsampled to 40MHz according to Nyquist Theorem while the narrower band signals is upsampled to the same level to ease processing. Then the sender adds the upsampled signals up at the mixer and delivers to the RF hardware, where it performs DAC and baseband to passband up-conversion to transmit the signals over the air. Since all signals have the same sampling rate regardless of their original data bandwidth, we can easily send them together. A similar technique has been used in SVL [32] and PICASSO [10].

Receiver: As shown in Figure 2, upon the arrival of a signal, the receiver performs passband to baseband down-conversion and ADC at the RF front-end and then uses the spectrum detector to determine which subband the sender uses for transmission. After the detector detects the start of the frame and estimates its channel width and center frequency, the receiver performs the following spectrum

shaping. First, if the RF center frequency and the receiving spectrum are different, similar to the center frequency shifting at the sender side, the receiver compensates it by multiplying $e^{-j2\pi\Delta n}$ to the received baseband signals. Second, it filters the signal outside the channel width by LPF to avoid the adjacent channel interference. Third, it downsamples the signal to match the sampling rate to that of the transmitter (*e.g.*, downsampling by a factor of $\frac{1}{2}$ for 10MHz signal). After that, the signal is delivered to the 802.11a PHY decoder. While the decoder is decoding the signal that has already been detected, the detector keeps sensing the spectrum to detect new signal from the remaining idle spectrum. The detail of the spectrum detector is introduced in Section 3.2.3.

A natural question in the receiver design is whether spectrum can be changed quickly enough to match that of the sender before the receiver starts channel estimation using the preamble. When the receiver changes its filter bandwidth, it incurs some delay since it generates valid output only after filling a few samples in the sample memory registers. This delay is called filter group delay [23], and is negligible compared to the preamble duration. As we will show in Section 3.2, our spectrum detection algorithm requires 64 samples while the preamble for detection consists of 160 samples. As most digital filters provide enough cutoff bandwidth when the group delay is less than 20 samples, the remaining preamble samples after the spectrum detection are enough for the adjusted filter to take effect before the channel estimation takes place. Moreover, the frequency shifting and downsampling incur close to 0 delay since they are simple to compute.

Upon detecting one or more transmissions and their spectrum profile, a receiver tries to decode all the detected frames and discard the ones that do not have its address as the destination. To further reduce processing overhead, one may optionally use PN sequence to encode receiver address so that the receiver can use simple cross correlation to detect if the frame is destined to itself (*i.e.*, a spike in the correlation between the received signal and it address encoded in PN sequence indicates the frame has its address). The receiver then only decodes the ones destined to itself, instead of attempting to decode all the frames.

3.2 Detecting The Spectrum

To achieve the per-frame spectrum adaptation, the receiver should know which center frequency and channel width the sender uses for transmission. One way to enable this is to coordinate the spectrum usage before communication (*e.g.*, using RTS/CTS exchange or control channel). Such coordination incurs significant overhead. Moreover, RTS/CTS exchange may not be possible without a shared channel between the sender and receiver. Another option is to let the receiver detect spectrum based on the characteristics of incoming signals. This is challenging because the flexible spectrum access not only allows the sender to freely change the center frequency and bandwidth but also lets it potentially transmit multiple signals concurrently. This significantly increases the number of ways in which spectrum is used.

Without loss of generality, this section assumes the center frequency of the RF hardware is 0MHz (*e.g.*, a 20MHz signal spans from -10MHz to 10MHz). Also, we assume that the center frequency of an incoming signal is discrete and set to maximize the number of narrow channels in a 20MHz channel (*i.e.*, 10MHz channel uses -5MHz and 5MHz as center frequency, and 5MHz channel uses -7.5MHz, -2.5MHz, 2.5MHz, and 7.5MHz as center frequency). This reduces the number of possible channels with little impact on spectrum efficiency. Below we first introduce 802.11a frame detection in Section 3.2.1. Then we develop a simple baseline scheme that uses temporal and spectral analysis for spectrum

detection in Section 3.2.2. In Section 3.2.3, we propose a novel spectrum detection method that exploits the 802.11a preamble signature in frequency domain.

3.2.1 802.11a frame detection

To give better understanding on the spectrum detection method, let us briefly explain how a normal 802.11 receiver detects the frame. The 802.11a preamble consists of Short Training Field (STF) and Long Training Field (LTF), where STF is used for the frame detection and LTF is used for the channel estimation. STF has a special property in time domain. It has ten periods, each with 16 samples, altogether 160 samples. The 802.11a receiver takes advantage of this periodicity to detect the preamble. It first measures the energy of the received signal r as $P = \sum_{k=1}^{L} r[k]r^*[k]$, where L is the number of signals used for energy measurement and * represents the complex conjugate of the signal. If P is higher than a given threshold, it performs auto-correlation check to see if the incoming signal is an 802.11a frame. The auto-correlation of the signal with d delayed samples C_d is:

$$C_d = \sum_{k=1}^{L} r[n+k]r^*[n+k-d].$$

The detection metric m_d is calculated as $m_d = \frac{C_d}{P}$ [9]. As the preamble has a period of 16 samples, the receiver uses m_{16} as the detection metric. If m_{16} is higher than the threshold γ, it considers a valid 802.11a frame is detected and starts further processing.

3.2.2 Spectral and Temporal analysis based spectrum detection (STD)

When the channel width of the transmitted signal is unknown to the receiver, it can be identified either by temporal analysis or spectral analysis of STF. Temporal analysis based detection detects the bandwidth of the signal using the characteristic of STF in the time domain. Specifically, the signal duration is inversely proportional to the bandwidth, so a narrowband signal lasts longer. For example, the duration of 20MHz and 10MHz signals are 50 ns and 100 ns, respectively. When a receiver using a larger bandwidth gets STF from a transmitter using a narrower band, it sees more samples every period. For example, STF on a 10MHz channel takes 1600 ns to transmit 16 repeated samples. A receiver using 20MHz will sample the signal every 50 ns and get 32 samples every period. Similarly, an incoming 5MHz STF signal yields 64 samples every period to the receiver using 20MHz. The temporal detection estimates the channel width using this property by measuring the number of samples in each period of a received STF. xGiven the received signal, the receiver measures m_{16}, m_{32}, and m_{64}. If all m_ks are below γ, the signal is considered as non-802.11a signal and ignored. Otherwise, it selects the channel width that gives the best match.

Although the temporal detection is useful for estimating the channel width, it is still insufficient since it cannot identify the center frequency of the signal. Moreover, it does not work when the center frequencies of the sender and receiver are not aligned. Spectral analysis can be used to detect both the center frequency and the channel width by measuring the energy of the signal in the frequency domain. The receiver calculates Power Spectral Density (PSD) by performing Fast Fourier Transform (FFT) of the received signal to identify both the channel width and center frequency.

However, this approach cannot distinguish how many transmissions exist in the detected spectrum when multiple transmissions continuously occupy the spectrum. For example, suppose that 10MHz spectrum is occupied by two 5MHz channel signals. Only relying

on the spectral analysis cannot tell whether it is one transmission on a 10MHz channel or two transmissions on two consecutive 5MHz channels. To address this issue, the temporal analysis and spectral analysis should be used jointly. When the spectral analysis detects spectrum is occupied from -5MHz to 5MHz, the temporal analysis is used to identify the exact channel width by comparing m_{32} and m_{64}. If both m_{64} and m_{32} are below γ, the signal is ignored. Otherwise, if m_{64} is higher than m_{32}, it considers the spectrum is occupied by two 5MHz signals; if not, it is considered as one 10MHz signal. This process is repeated for every possible spectrum combination in general and becomes computationally intensive when the sensed spectrum spans 20MHz due to many ways of occupying 20MHz spectrum.

3.2.3 Our Detection: Fine-grained Spectrum Detection (FSD)

In this section, we introduce a novel spectrum detection method that exploits the magnitude pattern of 802.11a STF in the frequency domain. The standard STF consists of 12 subcarriers with unit magnitude and 54 subcarriers with zero energy [19]. The interval between adjacent subcarriers with non-zero energy is 4 except some subcarriers near the guardband or the center, as shown in Figure 3 (a). We call such an interval as *subcarrier interval*. In Section 3.1, we mentioned that 5, 10, and 20MHz channel signals have different subcarrier widths. While the number of subcarriers is fixed to 64, the subcarrier width of 20MHz channel signal is four times and twice as that of 10MHz and 5MHz channel signals, respectively. We exploit this property to detect the channel width. Suppose a receiver whose channel width is tuned to 20MHz gets a signal from a sender using a 10MHz channel. Because the subcarrier width of the 20MHz receiver is twice as large as that of the 10 MHz sender, the signals of 2 subcarriers from 10MHz sender are merged into one subcarrier of 20MHz receiver, which reduces the subcarrier interval from 4 to 2 as shown in Figure 3 (b). Likewise, the subcarrier interval of 5MHz STF is 1 to the 20MHz receiver as shown Figure 3 (c).

(a)

(b) 10MHz   (c) 5MHz

(d) 5MHz 5MHz   (e) 10MHz 5MHz 5MHz
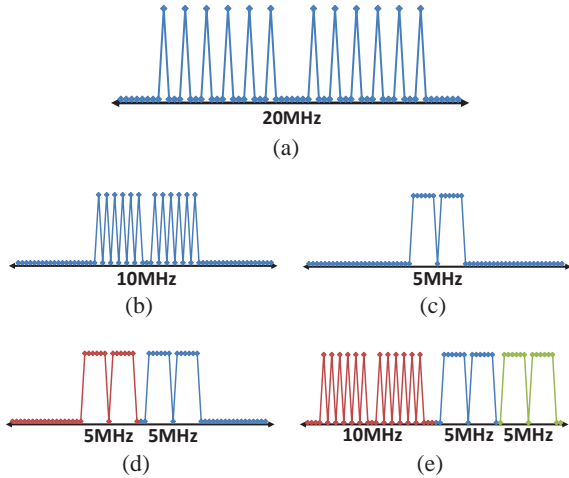
Figure 3: Different patterns of received STF in the frequency domain when the receiver uses 20MHz channel and the transmitter uses spectrum as indicated in the graphs.

These interval differences let us easily determine the channel width, center frequency, and number of channels. For example, comparing Figure 3(b) and (d), one can easily see the preamble pattern of one 10MHz channel transmission is visibly different from

that of the two 5MHz transmissions in the same spectrum. Depending on the combination of the channels that the sender chooses, it generates 29 different preamble patterns on a 20MHz spectrum.

ML detection algorithm: The above observation on the preamble patterns in the frequency domain transforms the spectrum detection problem into a pattern matching problem. As different channel combinations yield different preamble patterns at the subcarrier level, the detector can identify the spectrum by determining the pattern of the transmitted preamble. We develop a Maximum Likelihood (ML) detection algorithm. It is optimal when the probabilities of all of the channel combinations are equal. Let $\mathbf{Y} \in R^{64}$ be the magnitude vector of the received preamble in the frequency domain generated by performing FFT-64 on the received STF signals, and $\mathbf{X}_i$ be one of the possible preamble magnitude patterns generated by the channel combination ($i \in \{1, ..., 29\}$). Let $P(\mathbf{X}_i \mid \mathbf{Y})$ be the likelihood function that the transmitted preamble magnitude pattern is $\mathbf{X}_i$ given the received preamble $\mathbf{Y}$. We use the ML detection to find $\mathbf{X}_i$ that maximizes the likelihood:

$$\hat{\mathbf{X}} = \arg \max_i P(\mathbf{X}_i \mid \mathbf{Y})$$
$$= \arg \max_i \frac{P(\mathbf{Y} \mid \mathbf{X}_i) P(\mathbf{X}_i)}{P(\mathbf{Y})}$$
$$= \arg \max_i P(\mathbf{Y} \mid \mathbf{X}_i).$$

The second equality is according to Bayes Theorem, and the third equality is because $P(Y)$ is a constant and $P(X_i)$ is the same for all X_i since we do not have a priori that a certain magnitude pattern is more likely.

Next we show the ML detection essentially finds X_i that minimizes the Euclidean distance between the received signals and possible candidates across different subcarriers. Assuming Additive White Gaussian Noise (AWGN) channel and the probability distribution of all subcarriers are identical and independently distributed (i.i.d.),

$$P(\mathbf{Y} \mid \mathbf{X}_i) = \prod_{k=1}^{64} P(y^k \mid x_i^k) = \prod_{k=1}^{64} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y^k - x_i^k)^2}{2\sigma^2}},$$

where σ is noise variance of the channel and x_i^k and y^k represent the value of the kth element of $\mathbf{X}_i$ and $\mathbf{Y}$, respectively. As each transmitted subcarrier has either unit or zero magnitude, x_i^k is either 1 or 0. The result of the above ML detection does not change over the log-likelihood function. So we have

$$\mathbf{ln}P(\mathbf{Y} \mid \mathbf{X}_i) = \mathbf{ln}\frac{64}{\sqrt{2\pi\sigma^2}} - \sum_{k=1}^{64} \frac{(y^k - x_i^k)^2}{2\sigma^2}.$$

As a result, ignoring the invariants over i, the ML detection finds $\mathbf{X}_i$ that minimizes $\sum_{k=1}^{64} (y^k - x_i^k)^2$, which is the Euclidean distance between $\mathbf{Y}$ and $\mathbf{X}_i$. The detection algorithm optimizes the following objective:

$$\hat{\mathbf{X}} = \arg \min_i \sum_{k=1}^{64} (y^k - x_i^k)^2.$$

Binary detection to reduce the complexity: Our above Euclidean distance based detection can yield high accuracy, but calculating the Euclidean distance between the received magnitude vector and every possible candidate can be computationally costly. To reduce the computational cost, we also propose a binary detection method. It first classifies y^k to either 1 or 0 depending on its magnitude, and calculates $\sum_{k=1}^{64} (y^k - x_i^k)^2$. Because x_i^k and y^k are all binary numbers, $\mathbf{X}_i$ and $\mathbf{Y}$ can be represented as 64 bits sequences.

Then calculating $\sum_{k=1}^{64}(y^k - x_i^k)^2$ is simply counting the number of different bits between $\mathbf{X}_i$ and $\mathbf{Y}$ (*i.e.*, hamming distance). The resulting binary detection algorithm is represented as

$$\hat{\mathbf{X}} = \arg\min_i \mathbf{X}_i \oplus \mathbf{Y},$$

where $\oplus$ is bitwise XOR operation.

The complexity of the binary detection algorithm is $O(nk)$, where n is the number of subcarriers and k is the number of possible channel combinations. Specifically, in the proposed system design, n is 64 and k is 29. Due to simple XOR operation, this can be computed quickly. As shown in Section 5, the average processing time of the detection algorithm in SORA is $4\mu s$ with negligible variance. It will be much shorter if implemented in hardware chipset.

3.2.4 Discussion

Other WiFi standards: The spectrum detection above exploits specific structure in IEEE 802.11a preamble. One might wonder if this detection is applicable to the other 802.11 wireless standards. Our detection method is applicable as long as the detection preamble has inherently similar structure to the 802.11a STF in the frequency domain. Fortunately, IEEE 802.11n and 802.11ac maintain the same design of STF, so FSD is directly applicable to them. For example, 802.11n 40MHz STF has 128 subcarriers; among them, there are 24 subcarriers with unit magnitude with 4 subcarrier intervals [20]. Similarly for 80MHz and 160MHz STF in 802.11ac [21].

Now the remaining issue is if the algorithm is scalable enough for detection on a wider spectrum. Increasing channel width increases the number of ways in which spectrum can be used, and makes detection more costly. To address the issue, we can combine coarse energy detection with fine-grained spectrum detection. In the carrier sensing process, an 802.11ac device is required to separately perform CCA for every 20MHz channel. If the signal is detected on a 20MHz channel, it further performs FSD to find the subband usage pattern. If the energy of the 20MHz channel is below CCA threshold, FSD is not performed. In the worst case, the signal spans a 160MHz channel and spectrum detection on a 160MHz channel is close to eight times as that on a 20MHz channel since the detection cost is dominated by identifying how each of the eight 20MHz channel is used. We believe the increased detection cost is acceptable considering that 802.11ac devices are inherently more powerful than 802.11a devices in order to handle up to eight times sample rate, up to 8 spatial streams, and more complex modulation and channel coding, such as 256-QAM and LDPC [21].

Once the spectrum is detected, the receiver decodes the signal on the detected spectrum using the decoding scheme as specified by the standard. Namely, FSA does not affect decoding. For example, if MIMO is used, MIMO processing is evoked during the decoding as usual.

Detection accuracy: Note that one may be concerned that STF is too short for us to reliably extract frequency domain signal and cause noisy results since we do not know when STF starts. Fortunately, this is not an issue since we only rely on the magnitude pattern in the preamble for detection. As long as the error in detecting starting time is within cyclic prefix (CP) (*i.e.*, a guard interval in OFDM), this error only affects phase but not magnitude and will be cancelled out during channel compensation. Therefore, we can get correct magnitude of the received STF as long as the FFT-64 window resides within 160 STF samples, and hence our detection is robust even for short STF.

Antenna gain control (AGC): As shown in the previous section, the transmitted STF signal varies according to the channel width. Since a receiver performs AGC using the received STF, one might wonder if it affects AGC. Since it is difficult to find the exact starting sample of STF when it is detected, the exact pattern of STF is not utilized in AGC [13, 8]. Instead existing AGC algorithms control the hardware gain parameter using the running average of the received STF magnitude. Since the average magnitude of transmitted STF remains the same across different channel widths, different STF patterns have little impact on AGC.

Bidirectional traffic: When there is bi-directional traffic, FSA can be applied independently in both directions. Namely, node A can freely select the spectrum to transmit to node B and node B detects the spectrum in use based on the received preamble pattern and decodes the signals on the detected spectrum. Similarly, node B freely selects the spectrum to transmit to node A and node A also detects the spectrum using the preamble pattern and decodes the detected signals. If one of the nodes, say node A is an AP and communicates with multiple clients, the ACKs from the clients may overlap in time and the AP will detect the spectrum used by each ACK and decodes the ACKs in the order of their arrival time.

4. SPECTRUM ADAPTATION

4.1 Problem Specification

Given the capability of changing the spectrum on a per-frame basis, a natural question arises which spectrum should be assigned to a given transmission. In this paper, we consider the spectrum adaptation in an enterprise WLAN where all the APs are connected through an Ethernet and can be managed using a central controller. We focus on downlink traffic from APs to clients since downlink traffic often dominates the total traffic [16, 4].

While significant work has been done on channel assignment in WLAN, spectrum adaptation considered here differs from these existing works in the following important aspects. First, existing channel assignments assume an AP can transmit one frame at a time, whereas our flexible baseband design allows an AP to possibly transmit multiple frames simultaneously on different spectrum. Second, most channel assignment schemes (*e.g.*, [14, 17, 18, 28]) consider fixed channel width and center frequency. The flexible spectrum access enables us to dynamically change the channel width and center frequency and makes the channel assignment significantly different and have much larger search space. In particular, there can be an arbitrarily large numbers of channels in a given spectrum depending on the center frequency and channel width. Third, most existing channel assignments restrict an AP to use the same channel to communicate with all its clients for an extended duration of time. This is necessary due to significant overhead involved in changing the channels and also difficulties for the receivers to identify the spectrum used by the sender. [27] tries to alleviate the restrictions and consider flexible spectrum allocation. However, due to significant channel switching overhead (over 4ms) and the lack of per-frame spectrum detection at a receiver, its channel is adapted in a much longer time scale and it has to use multiple interfaces to hide the channel switching delay. Finally, our allocation problem differs from OFDMA scheduling (*e.g.*, [11]) in that OFDMA supports only synchronous transmissions while FSA can support asynchronous transmissions and allows a new mini-frame to start in the middle of other scheduled transmissions, which is common in asynchronous WiFi networks. Such an increased flexibility significantly increases the search space in the schedule and has the potential to yield better performance. It is feasible to use any part of the spectrum for a new transmission due to our fine-grained spectrum access and detection. Inspired by the existing works, we aim to completely remove all these restrictions and dy-

namically adapt the spectrum on a per-frame basis according to the instantaneous channel condition.

Our spectrum adaptation problem can be formulated as follows. Given a frame and SNR to the client across the entire spectrum, we decide which spectrum and start time should be assigned to the frame in order to optimize a given objective while ensuring interfering transmissions cannot overlap in time and spectrum. The objective we use in this paper is the finish time. In particular, we use the finish time of all transmissions as the primary objective; if this is equal, then we use the finish time of a new transmission to be scheduled as the secondary objective for tie-breaking when the primary objective is the same. Our algorithm can be easily extended to support other objectives.

4.2 Our Adaptation Algorithm

Challenge: The main challenge in the spectrum adaptation is the huge search space. Not only can a frame be placed in any part of the spectrum with different center frequency and channel width, but also a frame may not even occupy a continuous block of spectrum. This significantly increases the search space.

Joint spectrum adaptation and scheduling: In order to limit the search space, we use the following approach to discretize possible spectrum choices. We divide a frame into multiple mini-frames and divide the entire spectrum into multiple mini-channels. Our evaluation divides a frame with 1000-byte payload into 4 mini-frames, and uses 5MHz as the bandwidth of mini-channel. Then our goal is to determine where to place mini-frames (*i.e.*, which mini-channel and what is the starting time) to minimize finish time. A frame is allowed to occupy non-contiguous spectrum if needed.

We use the following greedy assignment with iterative improvement. For each incoming mini-frame, we first determine an initial assignment of the mini-channel and start time by going through all the mini-channels to find the one that optimizes our objective. Specifically, on each mini-channel we find the earliest start time such that the new mini-frame can transmit without interfering any existing scheduled transmissions while optimizing the finish time. We compute the finish time by measuring the SNR of the mini-channel and mapping the SNR to data rate according to the standard SNR to data rate mapping table (*e.g.*, the one in [25]). Next we go through each existing scheduled transmission to see if it is possible to swap the new transmission with one of them to further improve performance. If so, we swap with the existing transmission that gives the largest improvement. Otherwise, we stick to the initial assignment. Then we move on to the next mini-frame.

Figure 4(a) shows an example schedule, where (i, j) denotes the mini-frame that is in the i-th frame's j-th chuck. This 2-D schedule exploits the fact that our baseband allows an AP to transmit multiple frames simultaneously if needed. In comparison, the existing channel assignments, including latest ones (*e.g.*, [1, 27]), assume an AP can only transmit one frame at a time, thereby limiting the spectrum efficiency. Moreover, as it shows, different mini-frames do not synchronize in time due to our flexible baseband design.

Joint spectrum adaptation, scheduling, and AP assignment: So far, we focus on the spectrum adaptation and scheduling, and implicitly assume that clients have already been associated with an AP. However, there is inter-dependency between the spectrum adaptation and AP association. Simply assigning to the closest AP or the AP with the highest SNR is not always the best choice (*e.g.*, such an AP may be overloaded). Therefore, we propose a simple extension to support joint AP assignment, scheduling, and spectrum adaptation, which considers a 3-D schedule: (AP, mini-channel, time) as shown in Figure 4(b). It uses the same search strategy as the above except that now it searches over the 3-D dimension: which AP,
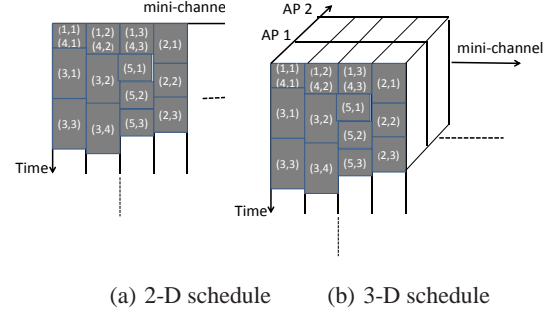

(a) 2-D schedule   (b) 3-D schedule

Figure 4: Example schedules, where (i, j) denotes the mini-frame from i-th frame j-th segment.

which mini-channel, and when to start in order to minimize finish time. As before, we first greedily find the earliest time to schedule the mini-frame and then use iterative improvement, which may swap the new mini-frame with one of the scheduled (but not yet transmitted) mini-frame if it improves performance.

Speed up: To enhance efficiency, for each mini-frame, we consider only a few top APs and a few top mini-channels in terms of SNR under no interference as candidate APs and channels, and prune the remaining APs and mini-channels from the search. Our evaluation considers the top 25% APs and mini-channel combinations. Moreover, when considering candidates for swap, only the transmissions that interfere with the new transmission are considered since otherwise swapping with it does not affect the new transmission's schedule.

5. TESTBED EXPERIMENTS

We implement FSA in SORA [33] on top of its IEEE 802.11a OFDM implementation. Our implementation includes the entire baseband system design specified in Section 3 based on User-Mode Extension (UMX) and PHY API provided in SORA SDK 1.6. We use DELL XPS 850 with Intel i7 Quad-core processor and 8GB memory for testbed setup. Without FPGA-level modification, we enable sampling rate conversion, center frequency shifting, concurrent transmission of multiple frames in different subbands, fine-grained spectrum detection, and decoding a single frame from subband in real-time. However, due to the capacity limit of the general purpose processor, we cannot decode multiple frames concurrently transmitted through multiple subbands in real time on SORA, so decoding is done offline using the captured signal dump as follow. The narrowband signal generated by the SORA node is transmitted over the air, and the receiver captures it using the signal capturing tool provided by SORA SDK under various channel conditions. The captured signals are fed to the SORA receiver to evaluate the spectrum detection and frame decoding performance offline. In some experiments, we use USRP2 devices to generate narrowband interference to understand its impact.

5.1 Spectrum Detection

One of the key techniques in FSA is spectrum detection scheme that allows fine-grained spectrum access without coordination with the receiver. To evaluate the spectrum detection performance, we place the transmitter and the receiver in various locations and measure the detection performance in varying channel conditions from 0 to 20 dB SNR. To demonstrate the advantage of FSA detection algorithm, we compare its performance with the temporal and spectral analysis based detection method introduced in Section 3.2.2 in terms of the detection accuracy and delay.
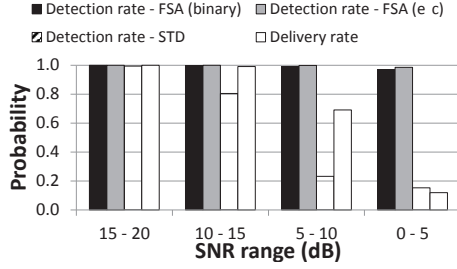
Figure 5: Spectrum detection accuracy comparison.

Figure 6: Detection accuracy under different channel widths.

Figure 7: Detection accuracy under different fading.

Detection accuracy: Figure 5 shows the detection accuracy under various SNR channels, where FSA and STD represent the proposed detection algorithm and spectral and temporal analysis based detection, respectively. We classify the measured results into 4 categories according to the SNR range, and plot the average detection probability in each range. Our testbed can run the proposed detection algorithm in real-time, but for fair comparison, we capture the received signal in the dump file and compare the detection performance using the same signal. To present the quality of the measured links, we plot the average frame delivery rate when the spectrum is known at the receiver and 6 Mbps rate Modulation and Coding Scheme (MCS) is used. Figure 5 shows that FSA achieves almost perfect detection performance. From 5 to 20 dB SNR range, FSA yields over 99% detection accuracy. In the severely degraded channel where the frame delivery rate is 10% (0 - 5 dB SNR in Figure 5), FSA still achieves 96% detection accuracy. The accuracy of FSA is comparable to that of the more computation intensive Euclidean Distance (ec) based detection, which is also our scheme and achieves accuracy of 98%. Therefore we focus on the binary detection due to its high accuracy and low computational cost in the remaining evaluation. In comparison, the detection accuracy of STD is unacceptable: when SNR is higher than 15 dB, it also shows close to 100% detection rate; but in 10 - 15 dB SNR range, its detection accuracy already drops to 80%, and when SNR is below 10 dB, its detection accuracy is only around 20%.

Why FSA and STD yield significantly different accuracy? FSA detects the spectrum using the magnitude patterns of the 64 subcarriers, which transforms the spectrum detection problem into pattern matching problem. With ML detection, it finds the pattern that gives the closest match. Assuming the binary magnitude, there are 2^{64} possible states of the subcarrier magnitude, but the number of possible preamble patterns is only 29, so the magnitude decision error of a few subcarriers is easily recovered in the ML detection process. The minimum hamming distance among the possible preamble patterns is 12. In coding theory, when the minimum hamming distance is d, the number of correctable errors is $\lfloor \frac{d-1}{2} \rfloor$ [30], which means FSA can tolerate mis-detection of 5 subcarriers magnitude. This implies that ideally our detection method is tolerable in a channel that gives 8% BER in BPSK modulation. On the other hand, the detection performance of STD relies on PSD analysis and temporal correlation check, which are sensitive to the channel noise. First of all, if PSD analysis mis-detects the allocated spectrum, there is no way to correct it. Moreover, when the estimated spectrum width is more than 10MHz, it performs temporal correlation comparison to identify the subband type, but this is unreliable in the low SNR channel.

Figure 6 evaluates the detection performance using various transmitted signal bandwidth in 0 - 20dB SNR channel. Increasing the spectrum width makes the detection more challenging because the power of STF is distributed over wider spectrum, which reduces the reliability, and a wider band also gives more possible subband
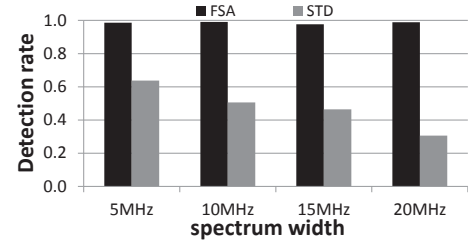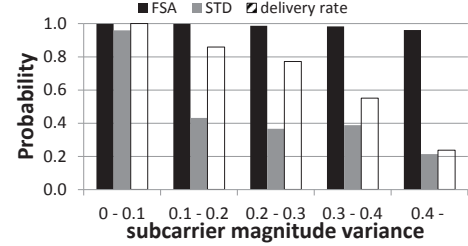
combinations. The result in Figure 6 shows that FSA is not affected by the signal bandwidth while STD yields lower detection accuracy as the bandwidth gets larger.

Figure 7 compares the performance in different fading channel. We measure the intensity of the fading using the magnitude variance of the subcarriers in the received frame. We place the nodes at different locations so that each link experiences distinct fading intensity, and measures the detection accuracy using the received signal capture. If the channel is flat, every subcarrier has similar magnitude and the variance is small. If the channel is frequency selective (*i.e.*, severe fading), the variance becomes large. Based on our observation, a variance of less than 0.1 indicates a flat channel, and a variance of larger than 0.4 indicates severe fading. Figure 7 is measured in channel with SNR 0 to 20dB. It shows STD only works in flat fading channel and its performance significantly degrades in medium level fading channel. On the other hand, FSA achieves over 97% detection accuracy even under severe fading.

Detection delay: For the real-time spectrum adaptation, the detection algorithm needs to be processed within the preamble time. In 802.11a, it should be done in 8 μs as the sampling rate and filtering parameters should be changed before LTF starts. Therefore, it is desirable that the detection algorithm is not only accurate but also computationally efficient. We compare the processing time of FSA and STD. This result is from our SORA testbed and it should be much shorter if it is implemented in the dedicated signal processing chipset, but this experiment demonstrates the relative complexity difference between FSA and STD. Figure 8 shows the cumulative distribution (CDF) of the detection delay when we measure the processing time of the detection algorithm 200 times for every
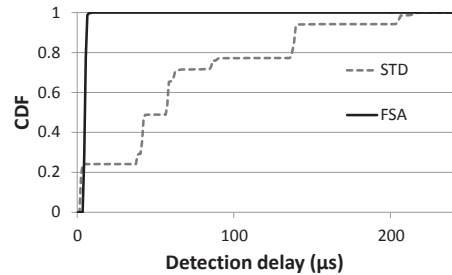

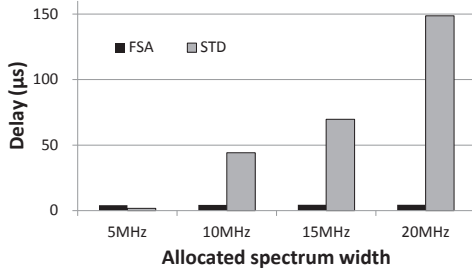**Figure 8: CDF of spectrum detection delay.**

Figure 9: Detection delay under different channel widths.

possible subband allocation. It shows that FSA significantly outperforms STD. The median delay of FSA is 4.2 μs, which enables real-time detection even in SORA implementation. On the other hand, the median delay of STD is 56 μs. Their 95th percentile delays are 5.6 μs and 204 μs, respectively. Moreover, the variance of FSA is only 0.44, whereas that of STD is 3218.

(a) without interference
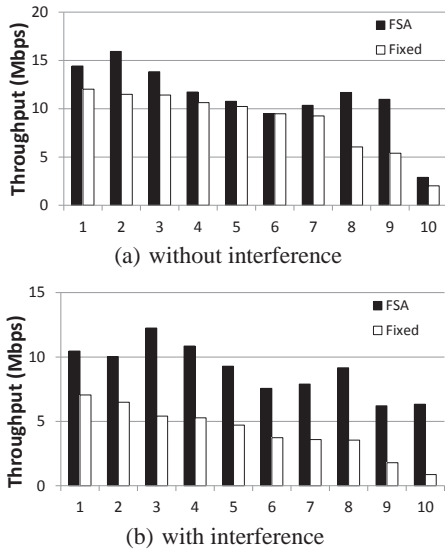

(b) with interference

Figure 10: Throughput comparison with and without narrowband interference.

The reason for the huge detection delay variance of STD is that its processing time depends on the detected subband width as shown in Figure 9. When the detected bandwidth is 5MHz, the spectral analysis can determine the subband type without further processing and achieves detection delay of 1 μs. However, if the detected bandwidth is more than 5MHz, it performs low pass filtering, frequency shifting, and auto-correlation check to identify the subband type. When 20MHz channel width is detected, it iteratively performs filtering and frequency shifting several times to check many possibilities, which takes 149 μs to process on average. On the other hand, FSA has almost constant computational cost regardless of the detected bandwidth, so its variance is small, which simplifies hardware implementation.

5.2 Spectrum Adaptation

Next we evaluate the performance gain of the fine-grained spectrum adaptation. We implement the spectrum allocation algorithm in Section 4 and compare its throughput with the fixed channel width approach. The channel bandwidth is 20MHz and the devices use flexible subband width from 5MHz to 20MHz as specified in Section 3. The AP selects the transmission rate from 6Mbps to 24Mbps based on the SNR feedback from the clients. We use 1

APs and 3 clients due to the limit of number of SORA nodes we have. When the AP allocates the spectrum and transmits the signal, the clients capture them. We evaluate the throughput by decoding them offline. We further evaluate larger scale networks using simulation in Section 6.

The testbed experiments evaluate the impact of the frequency diversity and how well FSA copes with the narrowband interference. For the latter case, we use a USRP2 device to generate 2MHz narrowband signal to interfere with some of the links in the testbed. We vary the location of the AP and clients, and report the average throughput from 10 runs with different channel conditions.

Figure 10 compares the throughput of FSA and the fixed channel width approach (fixed). The results are sorted by the throughput of the fixed channel width approach. In Figure 10(a), there is no interference and the throughput gain comes from the frequency diversity. In Figure 10(b), there is narrowband interference and the performance gain of FSA comes from both interference mitigation and frequency diversity. In the experiment without interference, FSA achieves the throughput gain of 27%. Since our allocation lets the AP use the subbands with higher SNR to communicate with each client, it improves link quality and increases the data rate. This gain increases with the number of clients or the width of spectrum [25]. But even with 3 clients and 20MHz spectrum, the gain is already significant.

When the clients are affected by the narrowband interference, the throughput gain of FSA increases to 110%. Though the interference is only 2MHz, it significantly degrades SNR for the fixed bandwidth approach. FSA avoids using the subbands with interference while taking full advantage of the subbands with high SNR.

6. SIMULATION

Next we use trace-driven simulation to evaluate the performance of spectrum adaptation.

6.1 Simulation Methodology

We compare the following spectrum assignment schemes: (i) random channel assignment with the fixed channel width (20MHz) and center frequency (RANDOM), (ii) greedy channel assignment that assigns the AP to the channel with least interference based on AP-to-AP interference again with the fixed channel width and center frequency (GREEDY), (iii) per-frame spectrum adaptation to change center frequency but fixing the channel width to 20MHz (FSA-fixed), (iv) per-frame spectrum adaptation with variable channel width where the minimum channel width is 5MHz (FSA-variable), and (v) FLUID-like scheme with a switching interval of 1 ms and 6 ms (FLUID - 1 ms/6 ms). Like FSA, FLUID also supports flexible channel widths, but an AP can transmit only one frame on any spectrum at a time, cannot split a frame across non-contiguous spectrum, and cannot change the spectrum within 6 ms to avoid the channel switching overhead. We also use 1 ms as the channel switching interval to reflect faster hardware in the future. We call the scheme FLUID-like since we did not implement the exact search heuristic described in [27] but use the same greedy search heuristic as FSA so that the performance difference between FSA and FLUID is due to the flexibility of the spectrum adaptation but not the quality of specific search algorithm, which can be applied to improve both protocols. For each spectrum assignment scheme, we run it with the following two different AP assignments: (i) assigning a client to the AP with the highest SNR (2D scheduling) and (ii) joint AP assignment (3D scheduling).

We collect the RSS traces using SORA nodes and varying the sender and receiver's locations. In this way, we measure RSS from 50 different links, each with 1000 frames. The traces report SNR
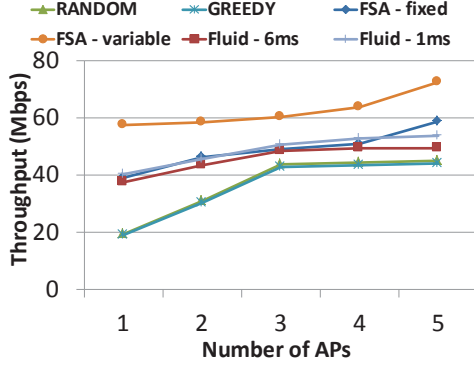
Figure 11: Vary the number of APs using static AP assignment (30 clients and 20MHz)

for every 5MHz channel across the entire 80MHz spectrum. We randomly map an AP-to-client link to one of the links in the traces. The SNR of the links ranges from 3dB to 20dB. We generate saturated traffic from the APs, record the finish time of different schemes, and calculate throughput as the total traffic transmitted divided by the time it takes. Transmission rate is selected as either 6Mbps, 12Mbps, or 24Mbps depending on the channel SNR.

In the simulator, each frame is divided into 4 mini-frames and a 20MHz channel is split into four 5MHz channels. In order to speed up FSA, we consider top 25% of mini-channels and APs in terms of SNR from the APs to the clients as candidate channels and APs.

6.2 Simulation Results

Varying the number of APs: Figure 11 compares different channel assignment schemes under the static AP assignment. We make several observations. First, as expected, FSA-variable > FSA-fixed ≈ FLUID > RANDOM and GREEDY. On average, FSA-variable out-performs static channel assignment (RANDOM and GREEDY) by 88% by adapting the spectrum per frame so that it can use the best spectrum for each transmission. FSA-variable out-performs FSA-fixed by 25% by leveraging fine-grained spectrum access instead of restricting to 20MHz channel. FSA-variable out-performs FLUID by 26% by supporting multiple transmissions, non-contiguous spectrum allocation, and more frequent spectrum adaptation. Second, as the number of APs increases, the throughput of all schemes improves quickly initially and then tapers off since the spectrum instead of the number of APs becomes the bottleneck when there are enough APs.

Figure 12(a) further compares different channel assignment schemes using joint AP assignment. FSA-variable > FSA-fixed > FLUID > RANDOM and GREEDY. On average, FSA-variable out-performs FSA-fixed by 47%, FLUID by 66%, and static channel assignment (RANDOM and GREEDY) by 155%. FSA-fixed out-performs FLUID since it can change center frequency more frequently whenever needed. Moreover, comparing Figure 11 and Figure 12(a), we observe that the benefit of joint AP and FSA-variable is 27% on average over its static AP assignment counterpart.

Varying the spectrum: Figure 12(b) compares various schemes as we vary the available spectrum from 20MHz to 80MHz. The relative ranking of various schemes stays the same. Moreover, as the spectrum becomes wider, the benefit of flexible channel adaptation increases due to more significant frequency diversity and narrowband interference in a wider spectrum. For example, FSA-variable out-performs the other schemes by 35% - 86% under one channel, and out-performs the others by 37% - 238% under four channels.

Varying the number of clients: Figure 12(c) compares various schemes as we vary the number of clients. As before, the flex-

ible channel assignment with joint AP assignment performs the best. On average, FSA-variable out-performs FSA-fixed by 43%, FLUID - 6 ms by 85%, FLUID - 1 ms by 64%, and RANDOM and GREEDY by 126%. We can see that the number of clients does not significantly affect the performance in all the schemes. This is because that once the scheduling queue becomes saturated, the number of frames that can be processed within the same time is almost fixed for a given spectrum assignment scheme regardless of the number of clients.

6.3 Comparison with WiFi-NC
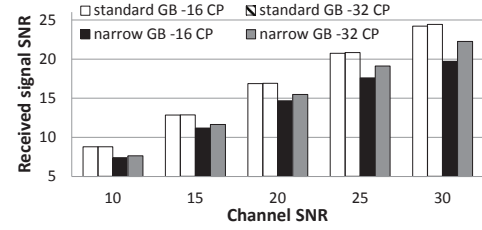
(a) RMS = $20ns$ (home environment)
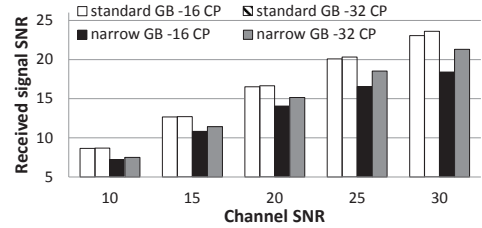
(b) RMS = $100ns$ (office environment)

Figure 13: Received signal SNR in various channel SNR, CP size, guard band width and RMS.

Next we use simulation to further compare FSA and WiFi-NC since the performance of WiFi-NC depends on the FFT configuration and Cyclic Prefix (CP). Similar to FSA, WiFi-NC also achieves the benefit of narrowband transmissions by dividing a channel into multiple subbands. The main difference is that WiFi-NC always uses a fixed narrow subband width while FSA uses variable widths. The downside of using variable channel width is that it requires an agreement between the sender and the receiver on the channel width. Section 5.1 shows we can accurately and efficiently detect the center frequency and channel width, so it is not a concern. A major limitation of the fixed narrow subband approach is that it needs more guard bands as the total channel width increases. To overcome it, WiFi-NC uses a very narrow guard band (*i.e.*, 100 KHz) and sharp cutoff filter. An inherent problem of the sharp cutoff filter is that it disperses symbols over time, which might increase inter symbol interference (ISI) [5, 23]. To enhance robustness against ISI, WiFi-NC uses a longer CP (*i.e.*, from 16 to 32) and increases FFT size from 64 to 128 to keep the ratio of data symbol to CP the same. However, this does not completely remove the overhead because (i) in some scenarios even CP with 32 samples is insufficient and can degrade SNR, and (ii) if we keep the CP size to 16 while increasing FFT size to 128, we can decrease the overhead ratio from $\frac{1}{4}$ to $\frac{1}{8}$.

In this section, we compare WiFi-NC and FSA using simulation. We first see the impact of the narrow guard band and sharp cutoff filtering in fading channel. Using MATLAB Signal Processing Toolbox, we design elliptic low pass filter with variable cutoff bandwidths. According to [5], we use 100KHz cutoff bandwidth filter generated by *ellipord* function in MATLAB. We model
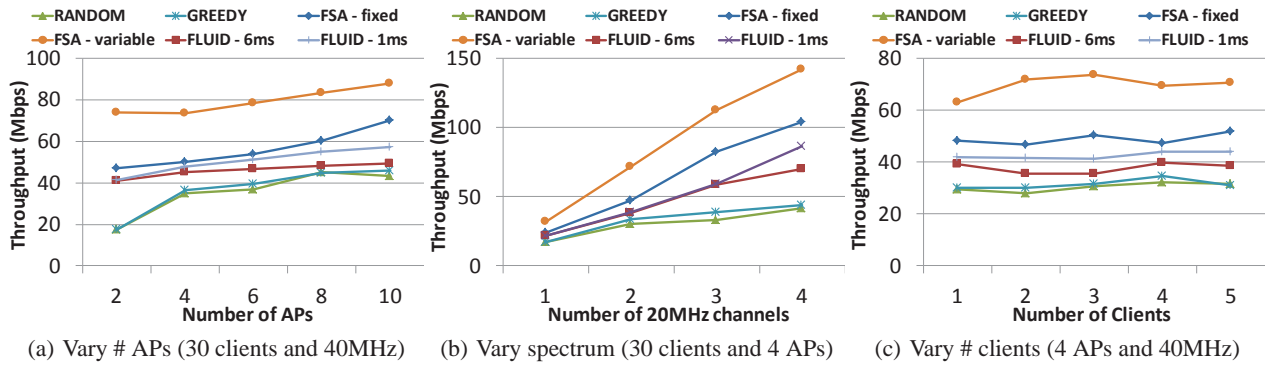
(a) Vary # APs (30 clients and 40MHz)   (b) Vary spectrum (30 clients and 4 APs)   (c) Vary # clients (4 APs and 40MHz)

Figure 12: Comparison of channel assignment schemes with joint AP assignment
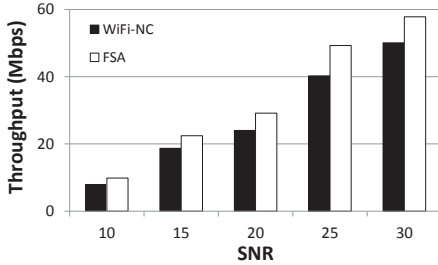

Figure 14: Throughput comparison between WiFi-NC and FSA.

the fading channel using Root Mean Square (RMS) of the delay spread. In home and office environment, RMSs are considered as 20 ns and 100 ns, respectively [22]. Figure 13 compares the quality of the received signal in various guard band width, CP length, and delay spreads. We divide a 20MHz channel into four 5MHz subbands, where the guard band width of WiFi-NC is 100KHz and the guard band width of 802.11a standard 5MHz is 712KHz [19], respectively. The received signal in each subband is filtered by elliptic filter where the cutoff bandwidth is set to the guard band width. In Figure 13, the received signal SNR is measured according to the error vector magnitude (EVM) [29]. CP sizes are set to 16 and 32. The result shows that in both home environment (RMS of 20 ns) and office environment (RMS of 100 ns), the quality of the signal with standard guard band is not affected by the CP size. As the CP size of 16 is enough to compensate ISI even in 100 ns RMS channel, increasing CP size does not improve the signal quality. On the other hand, the signal with narrow guard band experiences SNR reduction due to ISI from both multi-path fading and sharp filtering, and it is more serious when the CP size is 16. When the channel SNR is 30 dB and the RMS of the delay spread is 100 ns, SNR is 5dB lower than the signal with standard guard band in the same channel. A longer CP reduces the signal quality degradation, but even using a CP with 32 samples still yields 1-2 dB lower SNR than the signal with normal guard band, which suggests even a longer CP is required in this scenario.

Next we compare the throughput of WiFi-NC and FSA. We assume WiFi-NC and FSA use CP size of 32 and 16, respectively, while FFT size is fixed to 128. The channel bandwidth is assumed to be 40MHz, where FSA randomly divides it into 1 to 3 subbands. The guard band width of FSA is set to 600KHz, which is confirmed not to incur any SNR loss with CP size of 16 in 100 ns RMS channel. We assume that the frame aggregation is used so that WiFi-NC and FSA have the same MAC overhead. RMS of the channel is set to 20 ns to model the fading characteristic in home environment. Figure 14 shows the throughput in 5 different SNR channels. Here the transmission rate is selected as the one that gives the highest

throughput in the given channel condition. The result shows FSA achieves 20% higher average throughput than WiFi-NC. The gains come from the following factors. First, as WiFi-NC uses longer CP than FSA, it has larger constant overhead (around 5% more). Second, even with a longer CP, the signal quality is still lower than FSA with a smaller CP due to sharp filtering. This yields more errors and reduces throughput.

7. CONCLUSION

In this paper, we develop the first per-frame spectrum adaptation approach and prototype for WiFi networks. It consists of a flexible baseband design, a fast and accurate spectrum detection method, and an effective spectrum adaptation algorithm. Through implementation and simulation, we show this fine-grained spectrum adaptation significantly out-performs the existing approaches. It can also be incrementally deployed by adding APs with this capability while supporting legacy clients and APs. As part of our future work, we plan to conduct field trials to further understand and improve its performance in different environments.

8. REFERENCES

[1] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with WiFi like connectivity. In *Proc. of ACM SIGCOMM*, Aug. 2009.

[2] E. Chai, K. G. Shin, J. Lee, S.-J. Lee, and R. Etkin. Building efficient spectrum-agile devices for dummies. In *Proc. of ACM MobiCom*, 2012.

[3] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A case for adapting channel width in wireless networks. In *Proc. of ACM SIGCOMM*, Aug. 2008.

[4] Y. C. Cheng, J. Bellardo, P. Benk, A. C. Snoeren, and G. M. Voelker. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *Proc. of ACM SIGCOMM*, 2006.

[5] K. Chintalapudi, B. Radunovic, V. Balan, and M. Buettener. WiFi-NC: WiFi over narrow channels. In *Proc. of NSDI*, 2012.

[6] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *Proc. of ACM MOBICOM*, 2010.

[7] A. Cidon, K. Nagaraj, S. Katti, and P. Vishwanath. Flashback: Decoupled lightweight wireless control. In *Proc. of ACM SIGCOMM*, 2012.

[8] A. Fort and W. Eberle. Synchronization and AGC proposal for IEEE 802.11a burst OFDM systems. In *Proc. of IEEE GLOBECOM*, volume 3, pages 1335–1338. IEEE, 2003.

[9] J. Heiskala and J. Terry. *OFDM Wireless LANs: A Theoretical and Practical Guide*. SAMS, 2001.

[10] S. Hong, J. Mehlman, and S. Katti. Picasso: Flexible RF and spectrum slicing. In *Proc. of ACM SIGCOMM*, 2012.

[11] J. Huang, V. Subramanian, R. Berry, and R. Agrawal. In *Orthogonal Frequency Division Multiple Access Fundamentals and Applications*, chapter Scheduling and Resource Allocation in OFDMA Wireless Systems. Auerbach Publications, 2010.

[12] M. Jain, J. Choi, T. Kim, D. Bharadia, K. Srinivasan, P. Levis, S. Katti, P. Sinha, and S. Seth. Practical real-time full duplex wireless. In *Proc. of ACM MOBICOM*, 2011.

[13] V. P. G. Jimenez, M.-G. Garcia, F. G. Serrano, and A. G. Armada. Design and implementation of synchronization and AGC for OFDM-based WLAN receivers. *IEEE Transactions on Consumer Electronics*, 50(4):1016–1025, 2004.

[14] Y. Lee, K. Kim, and Y. Choi. Optimization of AP placement and channel assignment in wireless LANs. In *Proc. of IEEE LCN*, Nov. 2002.

[15] D. J. Love, R. W. H. Jr, , V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews. An overview of limited feedback in wireless communication systems. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, Oct. 2008.

[16] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks. In *Proc. of SIGCOMM*, 2006.

[17] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A Client-driven Approach for Channel Management in Wireless LANs. In *Proc. of IEEE INFOCOM*, Apr. 2006.

[18] A. Mishra, V. SHrivastava, D. Agarwal, S. Banerjee, and S. Ganguly. Distributed Channel Management in Uncoordinated Wireless Environments. In *Proc. of ACM MOBICOM*, Sept. 2006.

[19] L. S. C. of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard 802.11*, 2007.

[20] L. S. C. of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, amendment 5: Enhancement for higher throughput. *IEEE Standard 802.11n-2009*, 2009.

[21] L. S. C. of the IEEE Computer Society. Part 11: Wireless LAN medium access control (mac) and physical layer (phy) specifications: Amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz. *IEEE Standard 802.11ac-2013*, 2013.

[22] B. O'Hara and A. Petrick. *IEEE 802.11 Handbook: A Designer's Companion*. IEEE Press, 1999.

[23] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2009.

[24] Qualcomm. IEEE 802.11ac: The next evolution of Wi-Fi standards. *White paper*, May 2012.

[25] H. Rahul, F. Edalat, D. Katabi, and C. Sodini. FARA: Frequency-aware rate adaptation and MAC. In *Proc. of ACM MobiCom*, Sept. 2009.

[26] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat. Learning to share: Narrowband-friendly wideband wireless networks. In *Proc. of ACM SIGCOMM*, Aug. 2008.

[27] S. Rayanchu, V. Shrivastava, S. Baneree, and R. Chandra. FLUID: improving throughputs in entreprise wireless LANs through flexible channelization. In *Proc. of ACM MobiCom*, Sept. 2011.

[28] E. Rozner, Y. Mehta, A. Akella, and L. Qiu. Traffic-aware channel assignment in enterprise wireless LANs. In *Proc. of ICNP*, Oct. 2007.

[29] R. A. Shafik, M. S. Rahman, and A. R. Islam. On the extended relationships among EVM, BER and SNR as performance metrics. In *Proc. of International Conference on Electrical and Computer Engineering (ICECE)*, Dec. 2006.

[30] B. Sklar. *Digital Communications: Fundamentals and Applications (2nd Edition)*. Prentice Hall, 2001.

[31] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang. Fine grained channel access in wireless LANs. In *Proc. of ACM SIGCOMM*, Aug. 2010.

[32] K. Tan, H. Shen, J. Zhang, and Y. Zhang. Enable flexible spectrum access with spectrum virtualization. In *Proc. of Dyspan*, Sept. 2012.

[33] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker. SORA: high performance software radio using general purpose multi-core processors. In *Proc. of NSDI*, April 2009.

[34] L. Yang, W. Hou, L. Cao, B. Zhao, and H. Zheng. Supporting demanding wireless applications with frequency-agile radios. In *Proc. of NSDI*, April 2010.

[35] X. Zhang and K. G. Shin. Adaptive subcarrier nulling: Enabling partial spectrum sharing in wireless LANs. In *Proc. of IEEE ICNP*, 2011.