A Case for Adapting Channel Width in Wireless Networks

Ranveer Chandra, Ratul Mahajan, Thomas Moscibroda, Ramya Raghavendra[†], Paramvir Bahl
Microsoft Research, Redmond, WA [†]University of California Santa Barbara, CA
{ranveer,ratul,moscitho,bahl}@microsoft.com ramya@cs.ucsb.edu

ABSTRACT

We study a fundamental yet under-explored facet in wireless communication – the width of the spectrum over which transmitters spread their signals, or the channel width. Through detailed measurements in controlled and live environments, and using only commodity 802.11 hardware, we first quantify the impact of channel width on throughput, range, and power consumption. Taken together, our findings make a strong case for wireless systems that adapt channel width. Such adaptation brings unique benefits. For instance, when the throughput required is low, moving to a narrower channel increases range *and* reduces power consumption; in fixed-width systems, these two quantities are always in conflict.

We then present SampleWidth, a channel width adaptation algorithm for the base case of two communicating nodes. This algorithm is based on a simple search process that builds on top of existing techniques for adapting modulation. Per specified policy, it can maximize throughput or minimize power consumption. Evaluation using a prototype implementation shows that SampleWidth correctly identities the optimal width under a range of scenarios. In our experiments with mobility, it increases throughput by more than 60% compared to the best fixed-width configuration.

Categories and Subject Descriptors:
C.2.1 [Computer-Communication Network]: Wireless

General Terms: Measurement, Performance

Keywords: Channel width, spectrum, Wi-Fi

1. INTRODUCTION

Most wireless communication today involves the use of channels with preset widths. A wireless channel is the frequency spectrum block over which nodes transmit; it is uniquely specified by its center frequency and width. The use of preset channel widths is a direct result of how the available spectrum is divided by existing wireless technologies. For example, in 802.11 (Wi-Fi) b/g, the spectrum block is divided into 11 overlapping channels that are 20 MHz each and are separated by 5 MHz. Wi-Fi nodes communicate over one of these channels. In some cases, such as WiMax, the spectrum block is divided into channels of different widths. But even there the channel width is statically assigned.

In this paper, we argue that nodes in Wi-Fi networks should adapt the width of the communication channel based on their current needs and environmental conditions. To our knowledge, such adaptation has not been proposed or explored before. We find it surprising that Wi-Fi nodes dynamically change many variables today to improve communication, such as center frequency, transmission power, and modulation, except one of the most fundamental variable — the channel width.

We make our case in three steps. First, using measurements from controlled and live environments, we study properties of different channel widths. We use commodity Wi-Fi hardware manufactured by Atheros and make software modifications that lets these NICs to communicate at 5, 10, and 40 MHz channels in addition to the standard 20 MHz. We find that different widths perform differently on many measures of interest. Narrower channels have lower throughput but they have longer range, are more resilient to multipath delay spread, and consume less power. While these properties are broadly expected based on how our NICs implement different widths, our measurement study provides a detailed and systematic quantification. Actual and expected behaviors can differ quite a bit for commodity wireless hardware [5].

In the second step, based on our findings, we identify several unique benefits of dynamically changing channel width that are otherwise not available today. For instance, in times of low throughput requirement, nodes can simultaneously increase range and reduce power; in fixed-width systems, these two highly desirable properties are perennially in conflict. Another example is that total network capacity may be increased without increasing spectrum usage, by splitting multiple flows that share a wide channel into narrower channels. Yet another example is that nodes can substantially improve throughput by adapting channel width, because different widths offer the best throughput in different conditions.

Realizing these benefits requires practical channel width adaptation algorithms; in the third and final step, we show that this task is feasible at least in certain settings. We design a channel width adaptation algorithm, called SampleWidth, for the base case of two communicating nodes. It enables the nodes to adapt to optimize the throughput or power consumption of their communication. For efficient search and sampling, SampleWidth builds on top of existing techniques for adapting modulation. We present analysis and empirical evidence that its search converges to the optimal width.

We have prototyped SampleWidth on top of the same Atheros Wi-Fi NICs. Our experiments show that SampleWidth's simple adaptation scheme correctly approximates the optimal width in a range of distances between the sender and receiver. Even after including all sampling and channel switching related overheads, it stays within 10% of the optimal. In our mobile experiment, SampleWidth improves throughput by more than 60% compared to the best fixed-width system.

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

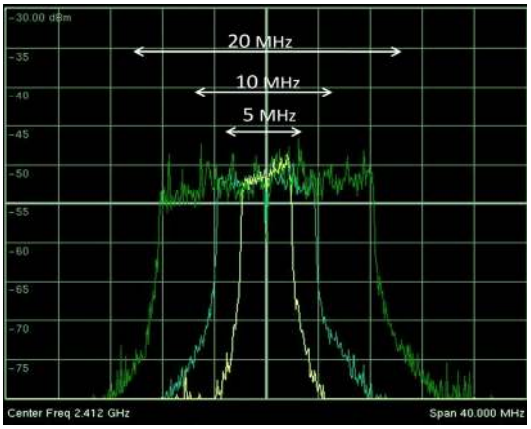


Figure 1: Screenshot of the spectrum analyzer showing 20MHz, 10MHz and 5MHz signals.

2. CHANGING CHANNEL WIDTH

We use the following terminology throughout this paper:

Channel width: The width of the spectrum over which the radio transmits (and receives) its signals; specified in MHz.

Throughput: Number of data bytes transmitted per second, including MAC-layer headers; specified in Mbps. We avoid the term bandwidth in this paper, as this term is frequently used to refer to both channel width and throughput.

Modulation: The specific modulation used by the radio while transmitting. We restrict our analysis to 802.11-based OFDM modulations that give data rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps when the channel width is 20 MHz.

2.1 Methodology

In this section, we describe the details of how we achieved different channel widths. The channel width of a wireless card is determined by the frequency synthesizer in the Radio Frequency (RF) front end circuitry. In most wireless systems, the frequency synthesizer is implemented using a Phase Locked Loop (PLL). A frequency divider on the PLL feedback path determines the center frequency of the card, and the reference clock frequency used by the PLL determines the channel width. Beyond this very high level description, we refer to [14] for details on the RF front end design of a wireless card.

We varied the channel width by changing the frequency of the reference clock that drives the PLL. We implemented this technique on off-the-shelf Atheros-based NICs. These cards use a clock frequency of 20 MHz to generate a 20 MHz wide signal. The value of the clock frequency can be configured in multiples of 2 using a hardware register. We changed the register values to generate signals on four channel widths of 5, 10, 20, and 40 MHz.[1]

We note that most Wi-Fi chipset designs, including Atheros, use a common reference clock for the RF transceiver and the baseband/MAC processor [4, 13, 19]. The baseband/MAC processor uses the reference clock to control access to the wireless network by regulating timing, encryption, encoding/decoding, and data transmission. Therefore, slowing or increasing the clock rate affects 802.11 timing parameters. For example, a 4 μs OFDM symbol in 20 MHz channel width gives symbols of length 2 μs in 40 MHz, and 16 μs in 5 MHz. Similarly, a 400 ns OFDM guard interval at 40 MHz is 3.2 μs at 5 MHz. We list a few important parameters that have different values at different channel widths in Table 1. We note that only timing parameters are affected. Therefore, irrespective of channel width, modulation 24 OFDM coding (24 Mbps at 20 MHz using 16-QAM, 1/2 rate encoder) carries the same 96 data bits per symbol. However, since symbol lengths are different across channel widths, modulation 24 coding scheme gives 6 Mbps at 5 MHz, 12 Mbps at 10 MHz, 24 Mbps at 20 MHz, and 48 Mbps at 40 MHz.

[1]Our 40 MHz channel width implementation is different from Atheros Turbo/SuperG mode. See Section 8 for a detailed discussion.

	5 MHz	10 MHz	20 MHz	40 MHz
Symbol Duration	16 μs	8 μs	4 μs	2 μs
SIFS	40 μs	20 μs	10 μs	5 μs
Slot Duration	20 μs	20 μs	20 μs	20 μs
Guard Interval	3.2 μs	1.6 μs	0.8 μs	0.4 μs

Table 1: A few 802.11 timing parameters across channel widths.

2.2 Implementation Details

All our changes are limited to the device driver software. The most important of these changes are as follows. We added a separate rate table with different rates supported by each channel width. The rate table is loaded by the driver when the channel width is changed. To ensure fair contention among flows on various channel widths, we modified the 802.11 slot time to be the same (20 μs) across all channel widths. The computation for packet durations were adjusted accordingly for different widths. For interoperability with 802.11b stations, 802.11g uses 4 802.11b DSSS rates (1, 2, 5.5 and 11), and 6 OFDM rates (12 to 54 Mbps), and uses a 44 MHz clock frequency. To isolate the impact of channel width, we modified the driver to use only OFDM rates (6 to 54 Mbps) in 802.11g mode. Also, for ease of exposition, we modified the clock frequency to 40 MHz so that channel widths scaled in multiples of 2. Finally, we added support to dynamically change the channel width without breaking 802.11 associations.

Figure 1 shows a spectrum analyzer screenshot on which different widths have been overlaid. It can be seen that while the center frequency for all widths during this measurement was 2412 MHz (corresponding to Channel 1 of IEEE 802.11 b/g), the channel width changes.

3. IMPACT OF CHANNEL WIDTH

In this section, we characterize the impact of channel widths on three of the key metrics of wireless communication: flow throughput, packet reception range, and power consumption. In all cases, we explain the underlying reason for the observed behavior and how it differs from what may be expected. The findings of this section form the basis of our arguments for dynamic adaptation of channel width.

Setup: For our experiments, we use two kinds of Atheros cards: i) Netgear WAG 511 (Atheros chipsets 5211 and 5212) which have a PCMCIA form factor for insertion into laptops; and ii) Netgear EnGenius' EMP-8602 modules, which are based on the Atheros 5213 chipset. These cards have a PCI form factor for insertion into desktops.

We performed experiments in a controlled emulator setup and in an indoor office environment. [2] We used CMU's wireless channel emulator [9], which has two laptops connected through an FPGA. The FPGA implements the digital signal processing (DSP) routines

[2]We also validated our results using an RF attenuator and outdoor experiments, but do not present results in this paper.

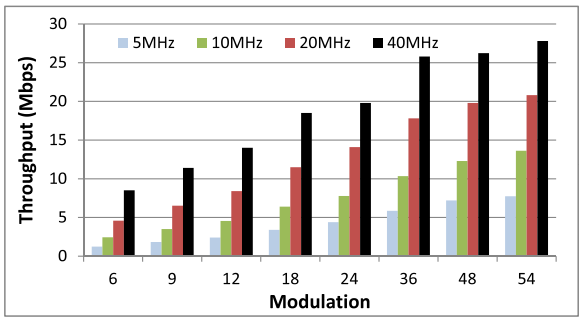


Figure 2: Impact of channel width on peak throughput of a UDP flow when packets are sent with different modulations.

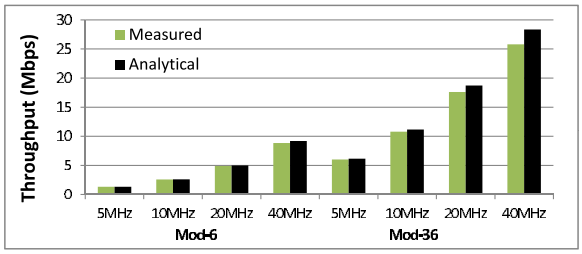


Figure 3: Actual throughput and model predictions for UDP traffic at different modulations for 5 and 40 MHz channels.

that model signal propagation effects, such as small scale fading and signal attenuation.

3.1 Peak Throughput

We start by understanding the impact of channel width on peak throughput of the communication. We measure peak throughput using the emulator to minimize the impact of external interference. In these experiments the signal is attenuated by only 20 dB. In other words, the receiver gets packets with good signal strength.

Figure 2 shows the throughput obtained by a UDP flow when using different channel widths and modulations. As expected, the throughput increases as the channel width or the modulation rate is increased.

According to Shannon's capacity formula the theoretical capacity of a communication channel is proportional to the channel width. Our measurements on commodity Atheros cards follow this relationship approximately but not exactly. The increase in throughput from doubling the channel width is less than a factor of two. For instance, at modulation 24, for 5 and 10 MHz the throughput is 4.04 and 7.65 respectively, which represents a factor of 1.89.

This less-than-doubling behavior is due to overheads in the 802.11 MAC, such as various inter-frame spacings. Since some of these overheads are fixed in terms of absolute time, e.g., the slot-time is 20 μs, their relative overhead for wider channels is higher. In order to more accurately capture the peak throughput at different modulations and channel widths, we extend the model presented in [18].

The idea of this model is to predict the time t_{packet} required for one single packet transaction. This total transaction time consists of SIFS, DIFS, and the time to send the data and the 802.11 acknowledgement: $t = t_{DIFS} + t_{data} + t_{SIFS} + t_{ack}$. The inverse of this per-packet transmission time multiplied by the number of bytes per packet exchange then corresponds to the throughput.

According to the 802.11g standard [1], the basic timing parameters in ad hoc mode are $t_{SIFS} = 10$ μs, $t_{slot} = 20$ μs, and $t_{DIFS} = 2t_{slot} + t_{SIFS} = 50$ μs. For the actual data packet, data is divided into a series of symbols, each encoding a number of bits. At modulation-R, $4 \cdot R$ data bits are encoded per symbol. The transmission time for each symbol is $t_{symb} = 4$ μs, and the data symbols are wrapped by a 20 μs preamble (synchronization and PLCP header) and a 6 μs signal extension.

To extend this model to adaptive channel width, we need to proportionally scale some of these timings with the width. Let B and R be the channel width and modulation, respectively, and let $\mathcal{B} = 20MHz/B$ be a scaling ratio. With the exception of the slot-time t_{slot}, all aforementioned timing parameters are scaled by the factor $\mathcal{B}$. Moreover, we discovered empirically (by varying t_{slot} and minimum contention window, CW_{min}) that the Atheros cards wait for an additional time of $t_{CW} = t_{slot} \cdot CW_{min}/2 = 8t_{slot}$ per packet. Therefore, putting all together, for a packet size of s bits, the time required for one single packet transaction is therefore

$$\begin{aligned} t &= t_{CW} + t_{DIFS} + t_{data} + t_{SIFS} + t_{ack} \\ &= 8t_{slot} + (2t_{slot} + \mathcal{B} \cdot t_{SIFS}) \\ &\quad + \mathcal{B} \cdot (20 + t_{symb}\lceil s_{data}/(4R)\rceil + 6) \\ &\quad + \mathcal{B} \cdot t_{SIFS} + \mathcal{B} \cdot (20 + t_{symb}\lceil s_{ack}/(4R_{ack})\rceil + 6) \end{aligned}$$

and $1/t_{total}$ exchanges per second can be completed. Multiplying by the number of user data bits per packet ($s_{data} - 76$ bytes$=$ 1460 bytes) yields the expected peak throughput.

In our setup, data and ACK size are $s_{data} = 1536$ bytes and $s_{ack} = 14$ bytes including all headers. R_{ack} is the rate at which the MAC-layer ACK packet is transmitted. In our setup, $R_{ack} = 6$ if $R = 6, 9, 12$, $R_{ack} = 12$ if $R = 18, 24$, and $R_{ack} = 24$ if $R \geq 36$.

Figure 3 shows how well the model predicts the throughput of the UDP flow at different configurations. At low data rates, our model almost exactly predicts the peak throughput for all four channel width options. The reason for the increasing discrepancy at wider channels and high modulations is that beacons and background noise (that are unaccounted for in the model) incur a higher per-packet overhead in these conditions.

3.2 Transmission Range

Changing the channel width impacts the transmission range of a wireless signal. This is primarily because of two main reasons: improved SNR and resilience to delay spread. We investigate the impact of both these factors on range in this subsection.

3.2.1 Improved SNR

We first investigate the resilience to noise using the emulator setup but unlike the previous experiment, we attenuate the signal between the two nodes. The transmission power of the radios is set to 1 mW, which is the minimum value supported by the chipsets we used in our experiment.

Figure 4 shows the loss rate as a function of the attenuation for different channel widths. The modulation is fixed to 6 in this graph. We see that narrower widths are able to withstand greater attenuation, which implies that they can reach further. We define the range threshold of the signal as the minimum attenuation at which the loss rate is less than 10%. Then, we can see this threshold is 74 dB for 40 MHz and 81 dB for 5 MHz. As we discuss below, this 7 dB difference is substantial because dB is a logarithmic unit.

The longer range of narrower widths can be explained as follows. For the same total energy used by a Wi-Fi radio to transmit a signal, the transmission power depends on the channel width measured in Hz, and power per unit Hz. Thus, at narrower widths, the radio can transmit with higher power per unit Hz without changing the total transmission power. Given equivalent noise per unit Hz across var-

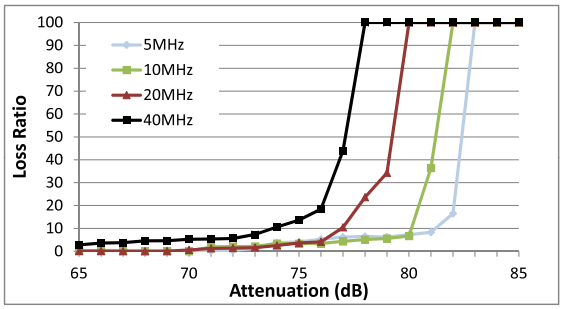


Figure 4: The loss rate as a function of attenuation for different channel widths at modulation 6.

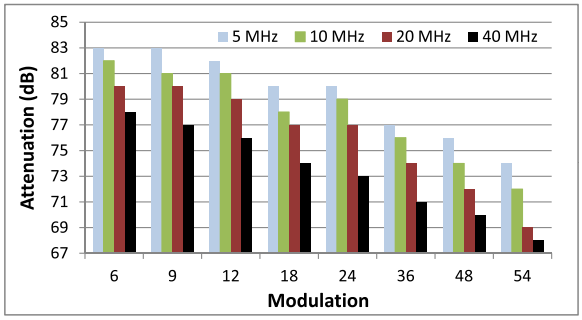


Figure 5: The range threshold for different channel widths and modulations. Higher threshold implies longer range.

ious widths, the SNR (signal-to-noise ratio) is higher for narrower widths, giving them a longer range.

However, the advantage we observe in practice differs from the maximum possible gain. As per above, halving the channel width should yield a 3 dB gain, or a 9 dB gain from 40 to 5 MHz. But the actual gain is only 7 dB across those two widths, which suggests that our hardware is leaving some potential gains on the table.

We repeated the experiment on an attenuator for different modulations. Figure 5 shows that the range advantage of narrower widths exists across all modulations. We see that lower modulations provide a range benefit that is almost equivalent to the emulator. Compared to 40 MHz at modulation 48, one can get a 6 dB range advantage either by reducing the channel width to 5 MHz while keeping the same modulation or by reducing the modulation to 12 while keeping the same channel width. One view, thus, of variable channel widths is that it offers finer scale modulations that otherwise do not exist.

To illustrate how the 7 dB advantage of 5 MHz over 40 MHz translates to better range in terms of real distance, we consider the following simplistic model. Assume that signal power decays as $1/d^{\alpha}$, with the distance d and path-loss exponent α, the maximum range A in dB attenuation corresponds to a maximum distance d_{max} as

$$A = 10 \log_{10} \left(\frac{P_{send}}{P_{recv}} \right) = 10\alpha \log_{10} d.$$

Therefore, we can estimate the proportional increase in range stemming from a ΔA dB increase in maximum attenuation (say, from A_1 to A_2) as $\frac{d_2}{d_1} = \frac{10^{A_2/(10\alpha)}}{10^{A_1/(10\alpha)}} = 10^{\Delta A/(10\alpha)}$.

α	2	3	4
range increase (est.)	123.9%	71.1%	49.6%

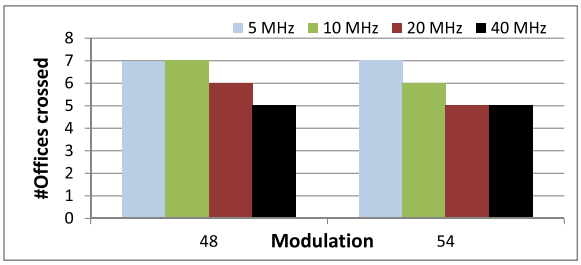


Figure 6: Indoor range for two modulations as a function of channel width.

The table above shows the range improvement as a function of α, which depends on the exact environment. Its value is 2 in free space and typically estimated as between 2 and 4 in real settings. The numbers above are meant as rough guidelines rather than precise predictions since we ignore multipath effects as well as many other practically relevant aspects of wireless signal propagation.

Figure 6 shows that the range benefits in reality roughly reflect the calculations above. In this experiment, we use an office as unit of distance and define range as the minimum number of offices crossed at which the loss rate between two nodes is 100%. This unit is of course very coarse but obstacles and severe multipath effects imply that exact signal attenuation is hard to quantify indoors. The offices are of identical size, and there are 8 offices in a straight line.

The graph shows results for modulations 48 and 54. At lower modulations, we could not reach the edge of communication for all channel widths. We see that narrower channels significantly increase range. At modulation 48, for instance, the range advantage of 5 MHz over 40 MHz is 3 additional offices or a 75% gain.

Finally, because an increase of X in range corresponds to an increase of X^2 in area covered, range increases can have significant practical impact for network coverage. Assuming a plane, for instance, the additional range in our indoor measurement amounts to over 200% more area.

3.2.2 Resilience to delay spread

At long communication distances, wireless receivers get multiple copies of a signal due to multipath reflections. Delay spread is the time difference between the arrival of the first and last copies of the multipath components. Delay spread can hinder correct decoding of a transmission at the receiver because a signal begins to interfere with a time-delayed copy of itself, also known as Intersymbol Interference (ISI). Modern radios use a RAKE receiver to counter delay spread, but their effectiveness depends on the coding scheme and the extent of delay spread [2].

OFDM specifies a *guard interval* at the start of every symbol to counter delay spread. For better packet recovery, a copy of the tail of the packet is included in the guard interval, called the cyclic prefix. For 802.11 at 20 MHz channel width, the guard interval is 800 ns, which is one-quarter of the symbol duration. This value of the guard interval has been shown to tolerate root-mean-square (r.m.s.) delay spreads of upto 250 ns [7]. Therefore, 20 MHz channel width provides good resilience to delay spread in most indoor environments, where the delay spread has been shown to be 50 ns in homes, 100 ns in offices, and 300 ns in industrial environments [8]. However, the delay spreads are larger in outdoor environments, even up to 1 μs, where IEEE 802.11 is known to give poor performance [2, 6].

As mentioned in Section 2, the guard interval increases by a factor of two each time the channel width is halved. Therefore, we expect higher delay spread resilience in narrower channel widths.

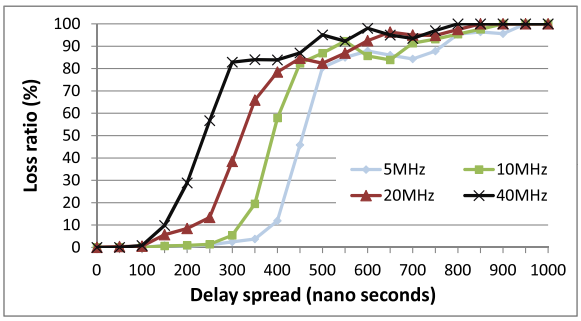


Figure 7: The loss rate experienced by different channel widths as a function of the delay spread configured in the emulator.

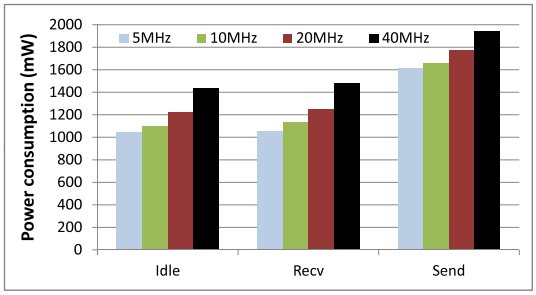


Figure 8: Power consumption of different channel widths in various modes.

To systematically evaluate the resilience of different widths to delay spread, we conducted controlled experiments using a wireless emulator. The emulator uses a two-ray channel model in which a delayed copy of the transmitted signal is attenuated and mixed with the original before arriving at the receiving radio. This emulates one direct line-of-sight signal and one reflected signal that followed a longer path. The parameters to this model are the delay between the two signals and their relative strengths at the receiver. In a real world setup, more reflected signals are likely to be present, but this experiment serves to provide an understanding of how channel width affects delay spread resilience.

For this experiment, we use the two-ray ground model in which the attenuation of the reflected ray with respect to the direct ray was set to 6 dB and the relative delay was input as a parameter. The direct ray was not attenuated. The delay spread is varied from 50 ns to 1 μs in steps of 50 ns and the broadcast loss rates between the laptops connected to the emulator is measured. Figure 7 shows the variation of loss rates with delay spread.

Figure 7 shows that narrower channels are more resilient to higher delay spreads. It plots as a function of the configured delay spread the loss rate of different channel widths. We see that 40 MHz is resilient upto about 150 ns delay spread, whereas 5 MHz can withstand about 400 ns. Based on the typical numbers above, we estimate that only 5 MHz is likely to work well outdoors.

3.3 Energy Consumption

We now quantify the effect of channel width on power consumption using a setup similar to the one used in [20]. We connect a 0.1 ohm resistor in series with the wireless card, and measure the current drawn through the resistor using a data acquisition system. We compute the power consumed by the wireless card by multiplying the current drawn through the resistor with the voltage supply of the wireless card (5 Volts).

Figure 8 shows the power consumed by different channel widths while idling, receiving, and sending packets. We present results for modulation 6, although, for the same channel width, the numbers were the same across different modulations. The figure indicates a *linear relationship* between the channel width and the power consumption. We see that wider channels consume more power. The additional consumption from 5 to 40 MHz is around 40% while idling and receiving packets and is 20% while sending packets. Thus, substantial powers savings can accrue from switching to narrower channels when appropriate.

The above measurements were conducted on the latest Atheros chipsets, AR5005GS, which have been optimized to consume less power when using a 20 MHz channel width. We also performed these experiments with older cards, off-the-shelf Netgear WAG511s, and the trend across bandwidths was similar, although the absolute numbers were much higher. For example, the power used to send was 2.17 W at 40 MHz channel width compared to 1.94 W with the newer cards. Similarly, the send power for 5 MHz channel width was 1.92 W instead of 1.61 W. We believe that further improvements in power profiles of Wi-Fi chipsets will lead to lower power consumption at narrower channel widths.

The decrease in power consumption can be explained by a slower clock speed that is used at narrower channel widths. In other areas of computing, energy optimization using clock frequency scaling of CPUs has of course been investigated for a long time, e.g. [10, 23]. Our results show that reducing the frequency of the clock in a Wi-Fi chipset also has a significant impact on energy consumption.

3.4 Results Summary

In summary, we showed the following properties:

• At small communication distances, throughput increases with channel width. The increase in not proportional to the channel width due to MAC layer overheads.

• Decreasing the channel width increases communication range. We get a 3 dB improvement by halving the channel width due to better SNR. Narrower channel widths also have better resilience to delay spread.

• Narrower channel widths consume less battery power when sending and receiving packets, as well as in the idle states. A 5 MHz channel width consumes 40% less power when idle, and 20% less power when sending packets than 40 MHz channel width.

4. BENEFITS OF ADAPTING WIDTH

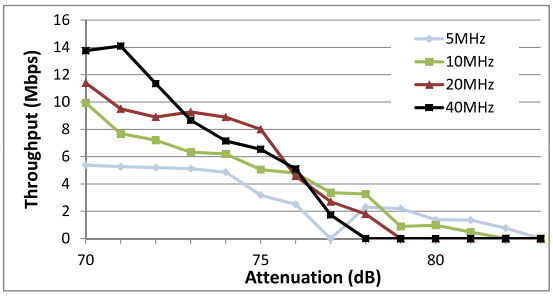
Having explored the basic capabilities provided by different channel widths, we now give some examples of how adapting channel width brings certain unique benefits.

A. Reduce power and increase range simultaneously

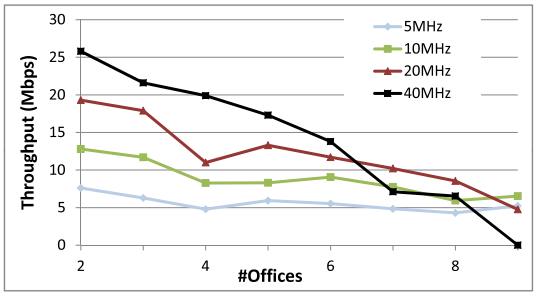
Fixed channel width systems face a tough choice between increasing range and reducing power consumption. They can increase range by increasing transmission power or using lower modulation. Using lower modulations does not change the instantaneous power consumption. Increasing transmit power increases battery power consumption. Adaptive channel width systems can have both! Narrower channels have both lower power consumption *and* longer range. Reducing channel width may come at the cost of reduced throughput, however, and so the width should be reduced when the additional throughput of the wider channel is not desired. Though, as our results below will show, in some cases narrower channels can improve throughput as well.

B. Improving flow throughput

The key motivation for our work is the following observation: although the peak throughput of wider channels is higher, the channel width offering the best throughput in a given setting depends on



(a) Emulator



(b) Indoor

Figure 9: Effective throughout offered by different channel widths at different attenuations and offices.

the "distance" between the nodes. We demonstrate this effect using emulator and indoor experiments.

Figure 9 (a) shows the effective throughput achieved between a sender and a receiver at different attenuations using autorate. Up to an attenuation of 72 dB, the highest throughput is achieved using the wide 40 MHz channel. In the ranges between 73–75 dB and 76–78 dB, it is best to use the 20 MHz and 10 MHz channels, respectively. Notice that the 3 dB optimality region for each of the intermediate channel widths (10 MHz and 20 MHz) exactly corresponds to the 3 dB range benefit predicted in Section 3.2.[3] Beyond 79 dB, the 5 MHz channel is the best choice. This throughput advantage of narrower channels stems from both their longer range and their ability to use modulations that are proportionally higher than narrower channels, after taking into account the inherent slowdown on narrow widths.

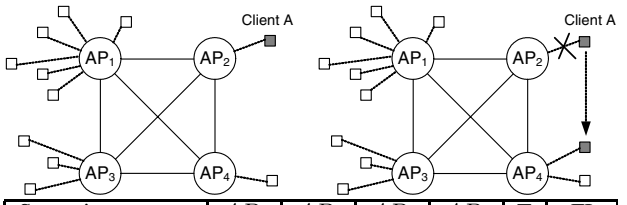
Figure 9 (b) shows the results from our indoor measurements. This experiment is limited by the fact that we do not have more than 9 offices in a straight line. But even within the extent to which we could measure, we can clearly see different offices (distances) have a different optimal channel width. While 40 MHz performs best up to the sixth office, 20 MHz outperforms all other channel widths in offices seven and eight. At office nine, 10 MHz is the best choice.

The crucial point is that there is no single channel width that serves all needs and hence, there is a strong case for adapting channel widths based on the current situation. In Section 5, we exploit these findings by designing a practical channel width adaptation algorithm that dynamically finds the best possible channel width.

C. Improving fairness and balancing load in WLANs

In today's 802.11g based WLANs, each AP is assigned a fixed width 20 MHz channel, and if possible, neighboring APs are placed on orthogonal frequencies. When the traffic is uniformly distributed

[3] As we mentioned before, because dB is a logarithmic unit, a 3 dB interval in which each channel width performs best maps to significant distance in real terms.



Scenario	AP_1	AP_2	AP_3	AP_4	T	FI
Case 1: (fixed)	1/6	1	1/3	1	4	0.58
Case 1: (adaptive)	2/6	1/2	1/3	1/2	4	**0.97**
Case 2: (fixed)	1/6	X	1/3	1/2	3	0.82
Case 2: (adaptive)	2/6	X	1/3	1/2	**4**	**0.97**

Figure 10: A network with four mutually interfering APs. With fixed channel widths are fixed, each AP is allocated a 20 MHz channel. In the adaptive scheme, AP_1 is allocated 40 MHz, AP_2 gets 20 MHz, , AP_3 and AP_4 get 10 MHz each. The tables shows the throughput received by each client after normalization by 20 MHz

across the network, such a scheme increases capacity and reduces interference. However, in dynamic conditions, using fixed-width channels can be problematic and suboptimal. Recent measurements have shown that there exists spatial and temporal disparity in client distributions [3, 16, 21] in large-scale WLANs. For example, a study of IBM's WLAN with 177 APs [3] showed that 40% of the APs never had more than 10 active clients, while a few APs in auditoriums and cafeterias had 30 simultaneous users; the set of heavily loaded APs also changes over time.

Adapting channel width of the APs offers a natural way to both improve flow fairness and balance load across APs. Consider Figure 4, which has four APs within interference range of one another. In Case 1 (left), AP_1 has 6 clients, AP_3 has 3 clients, while the remaining two APs have one client each. In Case 2 (right), client A moves away from AP_2 and associates to AP_4. We compare the performance of using fixed-width channels with adaptive-width channels. In the fixed-width channel case, the spectrum is divided into 4 channels of 20 MHz each. In the adaptive-width channel case, channels may be 10, 20, or 40 MHz. The table lists the throughput *per client* at each AP. Also included is the total throughput (T), and Jain's fairness index (FI). The index is calculated using $(\sum c_i)^2/n\sum c_i^2$, where c_i is the bandwidth obtained by client i, and n is the total number of clients.

In Case 1, the fixed-width channelization leads to severe unfairness among different clients. A client associated to AP_1 receives 1/6 of bandwidth compared to a client associated to AP_2 or AP_4. In contrast, with an allocation of 40 MHz to AP_1, 20 MHz to AP_2 and 10 MHz to the remaining APs, per-client fairness improves significantly to 0.97 because APs with many clients (AP_1) receive a wider part of the spectrum to serve its clients. Adaptive channel width can also help to improve system capacity. In Case 2, for instance, if client A moves from AP_2 to AP_4, an adaptive approach can reallocate the 10 MHz spectrum formerly used by AP_2 to AP_4, thus giving AP_4 a total of 20 MHz.

D. Improving network capacity

Many hardware and software improvements to wireless technologies are driven by the search for additional capacity. We present evidence that adapting channel width can provide another opportunity towards that goal. This benefit arises by partitioning conversations that share a wide channel into multiple narrower channels, which has the potential to increase capacity.

In this experiment, we use two sender-receiver pairs, i.e. four laptops. All four laptops were in communication range of each other, and we placed the two receivers close-by – two offices next

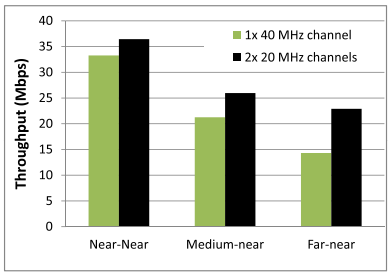


Figure 11: Average combined throughout of two flows when sharing a 40 MHz channel and when using adjacent 20 MHz channels.

to each other. We moved the senders to 24 different locations, and for simplicity present results for corresponding configurations in three categories. "Near-Near" is when both senders are within 3 offices of their receivers. "Medium-near" is when one sender is 4 or 5 offices away from its receiver, and the other sender is within 3 offices. "Far-near" is one sender is more than 5 offices from its receiver, while the other is within 3 offices.

Figure 11 shows the average combined throughput of the two flows when sharing one 40 MHz channel and when they are split on adjacent 20 MHz channels. We see that the gain is substantial – from 10% to 50% – in spite of any cross-channel leakage. The gain is maximum in the Far-near case because sharing the same channel introduces the rate anomaly problem by which the slower flow reduces total capacity. Separating the two flows on different channels lets the faster flow go faster. The other reasons for gain from splitting stems from reduced contention overhead and from the fact that narrower channels have a smaller per-packet relative overhead. We obtained similar results (not shown) when splitting two 20 MHz flows into adjacent 10 MHz channels.

We note that even though we do not increase total spectrum usage by splitting flows, we do increase total transmit power because narrower channels have higher power (although the same energy). It is thus an open question if the gain from such division persists in large-scale systems.

5. THE SampleWidth ALGORITHM

The previous section shows that substantial benefits can be had by dynamically adapting channel width. But realizing those benefits relies on practical adaptation algorithm. In this section, we present such an algorithm.

Our algorithm is called SampleWidth and it enables two nodes to dynamically select a channel width according to their workload and optimization criterion (e.g., throughput or energy consumption). This scenario forms the base case for channel width adaptation. It is of interest by itself in several settings: (i) two personal mobile devices (e.g., an iPod) sharing media content; (ii) a link in a multi-hop mesh network where the two nodes have a dedicated radio to talk to one another; and (iii) in 802.11 infrastructure networks where the AP has multiple radios on different widths and the client dynamically selects the best width. Besides, as we will show in this section, even this simple case has several intricacies that must be resolved before addressing the adaptation problem in more general settings.

5.1 Problem Definition

Consider two nodes, N_s and N_r. They have at their disposal k different channel widths $B_1, \ldots, B_k$. The goal of the algorithm is to select a channel width for a given objective. We assume that the two nodes have already decided which center frequency to use, for instance, based on their configuration or using some channel selection algorithm (e.g. [25]).

We consider two possible objectives in this paper, maximizing throughput from N_s to N_r, and minimizing the energy consumption of N_s. Simple extensions can optimize other measures including sum of the throughput or power across the two. We first describe our algorithm with the objective of maximizing throughput, or equivalently minimizing transmission time for a fixed-size transfer. In Section 5.5, we explain how the algorithm can be adjusted to minimize energy consumption.

5.2 Approach

One major challenge is the size of the search space. For a fixed transmission power, the main knob for optimizing transmission has been rate adaptation, i.e., finding the modulation that yields the best possible throughput. With the addition of variable channel width, the search space becomes two-dimensional. Even today, this represents 32 different options (8 modulations $\times$ 4 widths), and it may significantly grow in the future as more widths become available. Clearly, probing this entire search space is inefficient and we need methods that quickly converge to the optimal point.

However, we observe that the two dimensions can be decoupled. At any given width, to maximize throughput, the nodes must use the best possible rate. This problem of finding the best rate has been addressed by much previous work (e.g. [12, 15, 17]), which we leverage. SampleWidth uses a state-of-the-art autorate algorithm to find an efficient data rate on a specific width and then searches across widths. In addition to reducing the dimensionality of the search, this process enables us to search across widths less frequently and across rates more frequently. This is significant because in current hardware probing different channel widths incurs a coordination overhead, while searching across rates can be done on a per-packet level. To probe, both nodes are required to be using the same width.

Another source of overhead is the opportunity cost when probing suboptimal channel-widths. In the extreme case, if two nodes switch to a wider channel on which they are no longer within each other's range, they will disconnect and the subsequent reconnection may require significant time. Thus, sampling all widths is not practical, especially if more widths are available in the future.

To keep the cost of sampling low, SampleWidth is based on sampling only adjacent (i.e., the next narrower or wider) widths. It samples adjacent widths and switches if the sampled throughput is higher than the current throughput. Further, it probes the adjacent wider channel only if the probability of disconnection is low, i.e., if the average data rate on the current width is high. In Section 5.4, we show that this simple search strategy approach converges to the optimal channel width.

5.3 Algorithm

We now describe our algorithm in detail. In SampleWidth, nodes use the narrowest channel width when there is no data to send. This minimizes power consumption and increases the range, which is useful for mobile devices. Adaptation is triggered when there is data to send. Algorithm 1 provides a detailed description of the adaptation process. It proceeds in probing intervals of duration $t_S = 1$ s. The sender maintains a probing table with one entry for each available channel width, B_i, containing the average throughput T_i and average data rate R_i that autorate settled on when using this width. At the outset, all these entries are blank. During a probing interval, the sender measures the average throughput $\widehat{T}$ and data rate $\widehat{R}$ on the current channel width B_{cur}. At the end of

Algorithm 1 Channel-Width Adaptation Algorithm

Parameters: $\alpha = 9$ Mbps; $\beta = 18$ Mbps; $X = 5$;
$B_{cur} := B_1$;
[] —— *During each probing interval I do:*
Transmit using channel width B_{cur};
Measure avg. throughput $\widehat{T}$ and avg. data rate $\widehat{R}$;
[] —— *At the end of interval I do:*
Update probing table: $T_{cur} = \widehat{T}$; $R_{cur} = \widehat{R}$;
if $\widehat{R} \leq \alpha$ **and** B_{cur-1} has not been probed for X intervals
then
    Switch to next narrower width: $B_{cur} = B_{cur-1}$;
else if $\widehat{R} \geq \beta$ **and** B_{cur+1} has not been probed for X intervals
then
    Switch to next wider width: $B_{cur} = B_{cur+1}$;
else
    Find channel width B_ℓ for which $T_\ell = \max_{i=1,\ldots,|B|} T_i$;
    Switch to (or stay on) band B_ℓ: $B_{cur} = B_\ell$;
end if

the interval, it updates the corresponding entry in the probing table, and then, based on the most recent measurements and the state of the probing table, it decides whether to probe and switch to another channel width for the next probing interval.

This decision can be described using the two rules below. Rule 2 is executed if Rule 1 does not apply.

Rule 1a: If the current data rate $\widehat{R}$ is below a threshold α, the nodes switch to the adjacent narrower channel width. We argue in Section 5.4 that $\alpha = 9$ Mbps is optimal for current hardware.

Rule 1b: If the current data rate $\widehat{R}$ is above a threshold β, the nodes switch to adjacent wider channel. The optimal choice is $\beta = 18$ Mbps for current hardware. If the data rate at the current width is high, the probability of a disconnection when probing the next wider channel is low.

Rule 2: At the end of a probing interval, the nodes switch to the channel width B_ℓ for which the average throughput entry T_ℓ in the probing table is the highest.

To avoid oscillation, we slightly adjust Rules 1a and 1b such that the nodes do not probe a channel width if it was recently probed—within the last $X = 5$ probing intervals—and the throughput was lower than the current throughput.

Note that in SampleWidth, the decision to sample another width is based on the data rate, while throughput decides which channel width to use. This distinction is important because we cannot conclude from low throughput that moving to a wider channel is not beneficial. Low throughput can be caused by either poor link quality that causes many losses or high contention that creates fewer opportunities for transmitting. These causes need to be treated differently. In the first, probing and potentially moving to a narrower channel is the correct decision. In the second, moving to a narrower channel is unlikely to alleviate the problem. In fact, probing and possibly moving to a wider channel can help at high data rates.

The main advantage of limiting probing to adjacent channel widths is that only the most relevant channel widths are sampled. If nodes are using the currently optimal width of 10 MHz, for instance, SampleWidth may sample the adjacent widths (5 MHz and 20 MHz) depending on the achieved data rate, but unless conditions change (e.g., due to mobility), it does not waste time on sampling wider channels which are very likely to have poor performance. We now show below that limiting search to adjacent widths does not come at the cost of transmitting at suboptimal channel widths.

5.4 Optimality and Convergence

The critical question regarding the effectiveness of SampleWidth is whether it converges to the optimal channel width or gets stuck in a local minima. For instance, in a scenario where the optimal throughput is at 40 MHz but the throughput at 10 MHz is higher than at 20 MHz, nodes would be stuck at 10 MHz after starting at 5 MHz.

In order to formally show that such local minima are unlikely to exist, we introduce the notion of *smoothness* that captures the correlation between the channel width and the average data rate that autorate settles on.

For a given channel width B_i, let $R(B_i)$ be the average achieved data rate, i.e., the best data rate that autorate settles on when using channel width B_i. For a coarse approximation, let $T(B_i) = B \cdot R(B_i)$ denote the achieved throughput. No channel width should be a local minima. Formally, an intermediate channel width B_i is not a local minima if one of the two following properties holds for some constant $\lambda \geq 1$.

$$T(B_i) \leq T(B_{i+1}) \quad \Rightarrow \quad T(B_i) > \lambda \cdot T(B_{i-1})$$
$$T(B_i) \leq T(B_{i-1}) \quad \Rightarrow \quad T(B_i) > \lambda \cdot T(B_{i+1})$$

The constant λ quantifies the degree to which the above properties are satisfied. If at least one of the properties holds with $\lambda \geq 1$ then B_i is not a local minimum. Hence, for every channel width for which either $T(B_i) \leq T(B_{i+1})$ or $T(B_i) \leq T(B_{i-1})$, we define the smoothness of B_i as

$$\mathcal{S}(B_i) = \max\left\{\frac{T(B_i)}{T(B_{i-1})}, \frac{T(B_i)}{T(B_{i+1})}\right\}.$$

Based on this, we can define the smoothness of the entire system as $\mathcal{S} = \min_{B_i} S(B_i)$ over all channel widths that are not maxima.

The importance of smoothness stems from the fact that if $\mathcal{S} \geq 1$—i.e., if the system is convex— it guarantees that greedy local search converges to the global optimum. Moreover, if $\mathcal{S}$ is greater than 1, then the above properties becomes more robust and local search converges to the optimum point even if each sample may be inaccurate and even if autorate does not find the best possible data rate. Specifically in our case, if $\mathcal{S} > 1$, SampleWidth converges to the optimum even if the average data rate obtained by autorate is by a factor of $\mathcal{S}$ worse than the best possible modulation scheme. This is because for two channel widths B_i and B_{i+1}, the achieved throughput of such an autorate algorithm could deviate from the optimal $T(B_i)$ and $T(B_{i+1})$ by at most a factor of S, respectively. Hence, for smoothness $\mathcal{S}$, our algorithm can still decide which of the channel widths is better. On the other hand, if smoothness $\mathcal{S}$ is very low, less than 1, it implies that no efficient channel width adaptation algorithms exist because the optimal configuration can only be found if all options are sampled.

So, the question is, what is the value of $\mathcal{S}$? Intuitively, there are strong arguments why $\mathcal{S}$ should be at least 1. Our measurements in Section 3 show that the average data rate $R(B)$ is a non-increasing in B: as the channel becomes wider, the modulation yielding the best throughput drops. Importantly, our range experiments further indicate that once a critical attenuation is reached for a given channel width, the achievable throughput drops sharply. In the sequel, we capture this making only the very weak assumption that $R(B)$ is concave in B. For example, if the effective data rate is halved when going from 10 MHz to 20 MHz, it must drop at least as much when going to 40 MHz. If we thus assume that $R'(B) \leq 0$ and $R''(B) \leq 0$ holds[4], the second derivative of the resulting throughput function $T(B) = B \cdot R(B)$ is

[4]For the sake of simplicity, we assume B and $R(B)$ to be continuous for this argument. The same argument would hold for discrete values and difference quotients.

$\frac{\delta^2 T(B)}{\delta B^2} = BR''(B) + 2R'(B) \leq 0$, implying that $T(B)$ is a concave function in B. Hence, there is no local minimum and SampleWidth converges to the optimal channel width.

In theory, the wireless medium should thus be smooth even under minimal assumptions. Later, in Section 6.4, we empirically show that this is indeed the case even in our interference-ridden indoor setting.

Optimality of Parameters: Besides convergence, the other interesting question is regarding the choice of the two thresholds α and β. When determining the best possible values, we seek to satisfy all of the following objectives: i) avoid disconnections, ii) avoid unnecessary probing, and iii) probe new channel widths sufficiently often in order to avoid getting stuck on a suboptimal channel width. Clearly, these goals are contradictory. i) demands for a high value of β and ii) asks for low α and high β, respectively. On the other hand, in order to meet the third objective, the thresholds must not be too strict, i.e., not too low for α; not too high for β.

We determine the optimal values of α and β using our measured data sets. For a given setting (say, for a given distance or attenuation between sender and receiver), and for concrete values of α and β, we compute the long-term average throughput $T_{SW}(\alpha, \beta) = \lim_{t\to\infty} \frac{1}{t} \sum_t \widehat{T}_t$ that SampleWidth achieves when starting at some arbitrary width. Let T_{OPT} denote the average throughput achieved by a hypothetical optimal algorithm that constantly transmits using the best possible channel width. We can then define the *efficiency* $E_{SW}(\alpha, \beta)$ of a parameter pair (α, β) as the ratio between the throughput achieved by SampleWidth() and the optimum, $E_{SW}(\alpha, \beta) = T_{SW}(\alpha, \beta)/OPT$.

For each pair (α, β), we determined $E_{SW}(\alpha, \beta)$ based on our measurement numbers in the emulator and indoor experiments. As disconnections incur a particularly high cost, we discounted any pair of α and β that results in a disconnection. For all remaining pairs, we computed $E_{SW}(\alpha, \beta)$ for all attenuations (emulator) and all offices (indoor), and for all starting channel widths. Table 2 shows the computed values. It can be seen that our choice of $\alpha = 9$ and $\beta = 18$ provides optimal efficiency. Our choice of $\alpha = 9$ over $\alpha = 12$ is based on better average efficiency.

	$\beta = 12$	$\beta = 18$	$\beta = 24$	$\beta = 36$
$\alpha = 6$	0.20	0.20	0.20	0.20
$\alpha = 9$	0.47	0.94	0.70	0.66
$\alpha = 12$	0.47	0.94	0.70	0.66
$\alpha = 18$	0.47	0.91	0.69	0.63

Table 2: Efficiency $E_{SW}(\alpha, \beta)$ of SampleWidth for different values of α and β, and for $X = 5$.

5.5 Optimizing for Energy

The SampleWidth algorithm can easily be adjusted to optimize for other objectives. For instance, in order to minimize the power consumption of the sender (i.e., to pick the channel width that consumes the least power-per-bit), we only change the decision rule in Line 15. Instead of switching to the channel with highest throughput, we switch to the channel that is most energy-efficient. That is, we use EPJ_i instead of T_i to compare across different channel widths, where EPJ_i is the bits per Joule for channel width B_i. To compute EPJ_i for a sample interval, we need to know the number of bits successfully transmitted and the total energy spent. To compute the first term, we count the successful transmissions, and for the second, we also use packet retransmissions, the data rates used, along with the power numbers from Section 3.3 (see Figure 8). In general, these power consumption numbers may be different for different chipsets; we use the ones for our Atheros implementation. We show in Section 6.3 that the adjusted SampleWidth algorithm reduces energy consumption in comparison to fixed channel-width algorithms.

5.6 Implementation

Our implementation of SampleWidth is spread across user and kernel space as a daemon and a modified driver. Suitable hooks are provided in the driver to enable the daemon to issue adaptation commands. These hooks also enable the daemon to poll the driver for stats such as the current data rate and number of retries.

The daemon is responsible for initiating and maintaining the connection between the two nodes. The nodes send beacons periodically, containing information about their adaptation capability, and to advertise themselves to other nodes. When a node has data to send to another node that has been detected in range, the nodes form an ad hoc (peer-to-peer) network. When nodes connect and initiate a data session, the daemon initiates the adaptation policies, which in turn makes calls to the driver to switch the channel widths.

Because changing the channel width requires coordination between nodes (to ensure that both nodes are on the same channel width), we implement a simple handshake protocol. A node that wishes to change its channel width sends a request packet to the other node, and waits for an acknowledgement before switching the channel width. A node that receives a request packet switches the channel width right after sending the acknowledgement. In order to be robust against lost requests or acknowledgements, we implement a backup rendezvous protocol. If after changing the channel width, two nodes do not receive beacons for more than two seconds, they switch to the narrowest channel width and resume communication. In Section 6.5, we show that the overhead of switching widths is low in our implementation.

6. PERFORMANCE EVALUATION

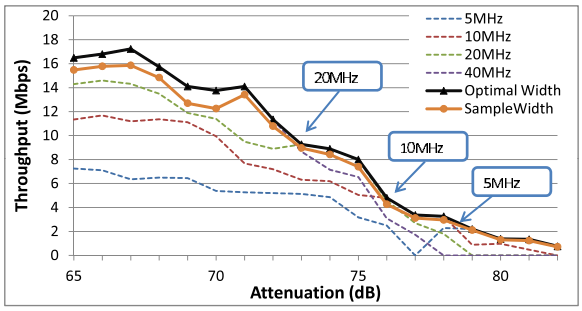
In this section, we evaluate SampleWidth along several dimensions. We will show the following.

- In Section 6.1, we show that SampleWidth approximates the throughput achieved by the optimal channel width for a range of distances and attenuations.
- In Section 6.2, using an experiment with mobility, we show that its adaptation to changing conditions lets it outperform the best fixed-width system by roughly 65%.
- In Section 6.3, we show how SampleWidth also saves power by selecting the most energy-efficient channel width depending on whether a data transfer is active.
- In Section 6.4, we show that current autorate algorithms come close to finding optimal modulation, and that the rate-width search space is sufficiently smooth to justify the use of autorate as a building block for SampleWidth.
- Finally, in Section 6.5, we show that the switching overhead of SampleWidth is small.

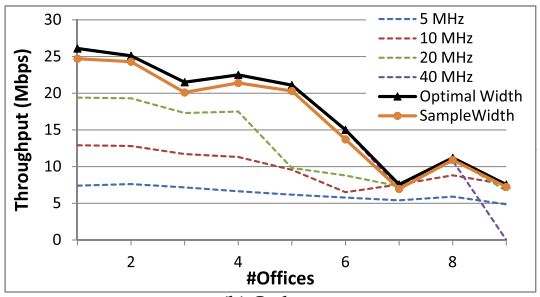
6.1 Choosing the Correct Channel Width

In this section, we evaluate how well the search strategy of SampleWidth is able to zero in on the optimal channel width. We consider throughput maximization as the objective and present results from both emulator and indoor experiments. In the emulator, we vary signal attenuation in steps of 1 dB and compute the UDP throughput for every available static channel width and then compare it to the throughput achieved by SampleWidth. The methodology is similar for the indoor environment except that the nodes are separated by varying number of offices.

Figure 12 shows our results for both settings, averaged over three



(a) Emulator. The labels depict transition points where that width becomes better that the adjacent wider channel.



(b) Indoor

Figure 12: Comparison of throughput achieved using SampleWidth with that of static width schemes in emulator and indoor settings.

runs. The plots show that for all attenuations and office distances the throughput achieved by SampleWidth closely tracks the throughput yielded by the optimal width. The maximum gap between SampleWidth and optimal throughput in the indoor experiments is only 8.7%, which includes all overheads stemming from probing adjacent widths as well as switching widths itself.

Width (MHz)	5	10	20	40	SampleWidth
Throughput	3.60	5.17	8.27	7.92	13.68

Table 3: Throughput achieved by fixed widths and SampleWidth in an indoor mobile network.

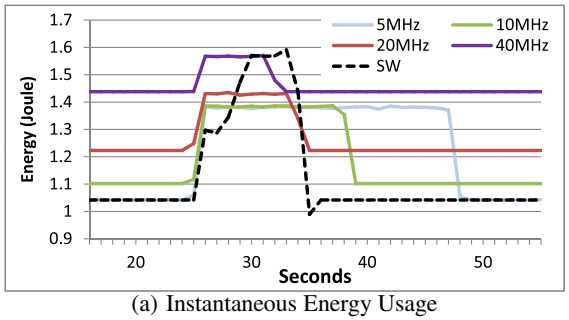
6.2 Adapting during Mobility

The previous experiment shows that SampleWidth adapts to the optimal width in stationary scenarios; we find that it is nimble enough to adapt well in mobile scenarios as well. We conduct a simple experiment in an indoor setting with a UDP transfer between two laptops. The receiver is positioned in a fixed location and the sender moves along a fixed trajectory at roughly constant speed. For different trials of this experiment, the laptops are either fixed on one channel width or use SampleWidth. Since estimating the optimal throughput in this setting is difficult, we evaluate SampleWidth by comparing it to the throughput of fixed-width scenarios.

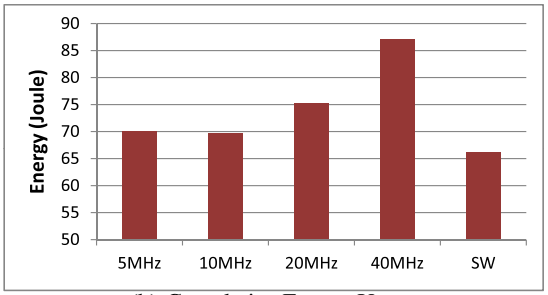
Table 3 shows the throughput of the fixed width configurations and of SampleWidth. We see that SampleWidth improves throughput by roughly 65% compared to the best fixed-width (20 MHz).

6.3 Reducing Power Consumption

We now evaluate the effectiveness of SampleWidth towards conserving power. In this experiment, each trial is one minute long and



(a) Instantaneous Energy Usage



(b) Cumulative Energy Usage

Figure 13: Instantaneous and cumulative energy usage for different configurations.

involves transferring a 20MB file 25 seconds into the experiment. We try all fixed widths and SampleWidth.

Figure 13(a) shows the power consumption behavior in detail for all configurations at the sender. The fixed width systems start out at their idle mode power consumption, move to their send mode consumption level, and then come back to their idle mode levels. SampleWidth starts out at the idle mode level for 5 MHz, because that is least costly. When the transfer starts, it moves to the power consumption level of 40 MHz, because that yields the least power-per-byte ratio. When the transfer finishes, it comes back to the 5 MHz level. Figure 13(b) shows that through this adaptation, SampleWidth is able to consume the least total amount of energy.

6.4 Efficiency of Autorate & Smoothness

SampleWidth uses autorate to probe channel widths and find an efficient data rate. We justify this design choice by showing that modern autorate algorithms are indeed capable of achieving close to optimal throughput. Figure 14 shows the suboptimality in terms of reduction in throughput of using Atheros's proprietary autorate implementation on Windows XP in comparison to using the best possible modulation in a stationary indoor setting. The important observation is that at all measurement points, autorate performs within at most 16% of the optimal data rate.

	1	2	3	4	5	6	7	8	9
$\mathcal{S}$	1.50	1.50	1.51	1.53	1.43	1.6	1.47	1.6	1.63

In order to see whether autorate is sufficiently close to the optimum in order for SampleWidth to converge, recall the definition of smoothness $\mathcal{S}$. Specifically, we have discussed in Section 5.4 that if the average data rate obtained by autorate is by no more than a factor of $\mathcal{S}$ worse than the optimum, SampleWidth is guaranteed to converge. Table 6.4 contains the $\mathcal{S}$ values of our indoor measurements. It shows that autorate is well within the required accuracy bounds indicated by these smoothness numbers and hence, it converges.

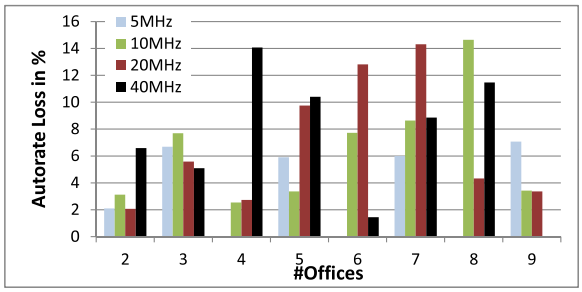


Figure 14: Suboptimality of autorate.

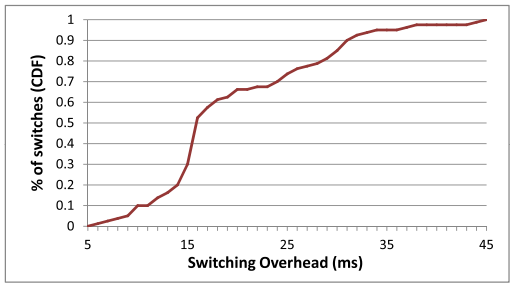
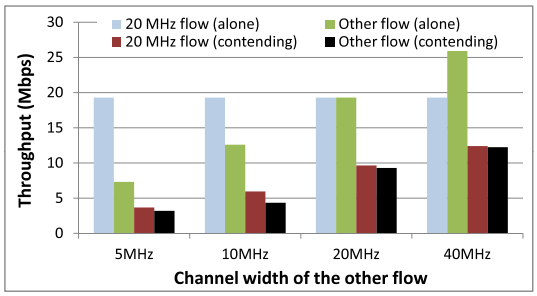
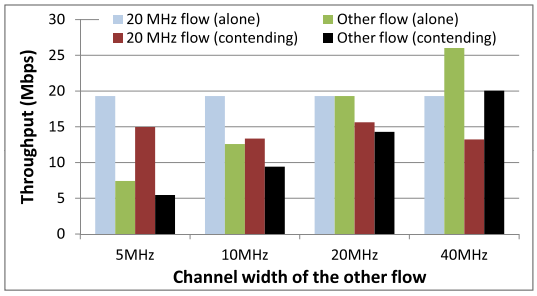


Figure 15: CDF of switching overheads when switching between a series of two random channel widths.



(a) Near setting



(b) Far setting

Figure 16: Maximum throughput at different office distances for all channel widths.

6.5 Switching Overhead

Finally, we quantify the overhead of switching widths in our current implementation. The setup consists of two laptops. One laptop broadcasts packets at a high rate, and also periodically coordinates with the other laptop and switches channel width. We measured the time elapsed at the receiver between when the ACK was sent and the next broadcast packet was received. That is, this time includes both the hardware switching time and the overhead of our coordination handshake, which is currently implemented in user space.

Figure 15 shows the CDF of switching overheads encountered for a series of random channel width changes. The results show that the median switching time is 17 ms and the maximum is 45 ms. These are small enough for most applications to not notice the underlying switch.

7. WIDTH INTEROPERABILITY

An important concern regarding what we propose in this paper is *cross-width interference*, i.e., interference between transmissions on different channel widths but the same center frequency. In today's Wi-Fi networks, nodes typically operate on orthogonal channels. The cross-channel interference is usually low, and nearby nodes on the same channel reduce simultaneous transmissions using a combination of physical and virtual carrier sensing (using network allocation vector, or NAV). In our proposed world, however, nodes will share overlapping frequency blocks, without being able to virtually carrier sense each other because they cannot decode each other's transmissions. If physical carrier sensing, which would still continue to function, is not sufficient to prevent nodes from trampling each others transmissions, the radio environment will become unusable.

While evaluating this aspect comprehensively is difficult within the scope of this paper, we present preliminary evidence that physical carrier sense may suffice to limit most simultaneous transmissions. We use two diverse settings, which we call "near" and "far". Both settings have two flows. In the *near* setting, all four nodes are in same office. In the *far* setting, the corresponding sender-receiver pair are in the same office, but the two pairs are separated such that they can only partially hear each other. The loss rate from the sender of one flow to the receiver of the second is around 50% at 20 MHz with Modulation 6. We measure the throughput of each flow when operating by itself and when operating with the second. In this experiments, one flow is always at 20 MHz. We vary the channel width of the other flow.

Figure 16 shows that flows co-exist well in both settings. In the *near* setting, when the other flow is at 5 or 10 MHz, the sum of the throughputs of the two flows does not add to the throughput of the 20 MHz flow alone, because of a version of the *rate anomaly* problem [11]. The rate anomaly problem refers to the case when a low modulation transmitter reduces the total throughput of the network. It occurs because the 802.11 MAC allows transmitters to send the same number of packets. But since the low modulation packets occupy the medium much longer, its presence reduces total network throughput. The same effect happens with narrower channels because their transmissions occupy the medium longer. These experiments suggest that different widths on overlapping spectrum blocks can co-exist.

8. RELATED WORK

The width of a wireless communication channel is one of the most important parameters in wireless communication. Surprisingly, fixed channel widths have been taken for granted in virtually all wireless networking research. In comparison, other knobs for improving network characteristics, such as transmission power, frequency assignment, or modulation have been investigated extensively. Our discussion in Section 4-B on load-balancing in infrastructure based WLAN networks exemplifies this point. While numerous approaches to alleviate this problem have been proposed based on power control, channel assignment, client-association, or rate-adaptation, channel width is always assumed to be constant.

Most recently, the wireless industry has begun exploring the use

of different (albeit static) channel widths. For example, the 2007 version of the IEEE 802.11 standard [1] specifies 5 and 10 MHz wide channels for use in the 4.9 GHz public safety bands. The WiMax [24] standard specifies 8 different channel widths mainly for compliance in international markets and to meet FCC regulations. Atheros has a proprietary Turbo mode, in which an AP can use 40 MHz wide channels if a client is turbo-mode capable. However, the allocation of channel width is static, i.e. either 20 or 40 MHz. Turbo mode does not operate in ad hoc mode. It is also known to be extremely unfair to legacy 20 MHz transmissions in its vicinity [22]. In this paper, we go beyond these exciting developments in the industry. We show that it is possible and beneficial to *adapt* the channel width based on application and system requirements. We also show how different bandwidths can co-exist without causing the unfairness of Atheros Turbo mode. In the realm of communication in cognitive radio networks over TV bands, our previous work on KNOWS [26] implicitly uses a notion of adaptive channelization. However, the work does not specifically explore practical benefits of adaptive channel width.

One technology to adapt spectrum utilization is Orthogonal Frequency Division Multiple Access (OFDMA) [24]. It is an extension of OFDM, in which different subcarriers within a fixed width symbol can be assigned to separate users. A user can be assigned non-contiguous subcarriers to improve resilience to narrowband interference. We note that OFDMA is complementary to our approach of changing channel widths. While we change the symbol duration to explicitly influence spectrum utilization, OFDMA can pack multiple users in the same symbol. Adaptive channel widths can give the benefits of throughput, capacity, range and power, while these benefits can be further enhanced using OFDMA technology.

9. CONCLUSIONS AND FUTURE AGENDA

In this paper, we demonstrate for the first time how—using standard, off-the-shelf hardware—the channel-width of IEEE 802.11-based network communication channels can be changed adaptively in software. Our measurements show that this can lead to significant improvements in many of the desirable metrics in wireless networks: range and connectivity, battery power-consumption, and capacity. This, in turn, indicates that using channel-width as a new, powerful tunable knob could lead to faster, less power-consuming, fairer, and ultimately better wireless networks.

Several hardware and software challenges must be met to fully realize the benefits of adapting channel width. On the hardware side, the most useful capability would be for radios to be able to decode packets at different widths (on the same center frequency). This capability would eliminate the coordination cost from channel width adaptation and allow nodes to unilaterally adjust width. The implementation of this capability could be similar to how radios can decode different modulations today: an initial header transmitted at a lowest width reveals the width of the remaining packet. In our experiments we observed leakage for narrower channels, perhaps because the hardware filter is designed for 20 MHz. A programmable filter for variables widths can reduce leakage and improve performance when adjacent narrow channels are used.

On the software side, the combination of variable channel widths and multiple center frequencies offers rich possibilities for improving system performance. Harnessing them requires new algorithms and models that are distinct from today's graph-coloring based fixed-width channel assignment models. The possibility of variable channel widths significantly changes the nature of the algorithmic problem that now must be cast as "interval-allocation." In addition, the varying capabilities of different channels, fragmentation concerns, and coordination cost must also be considered.

10. REFERENCES

[1] IEEE Std 802.11-2007 IEEE Standard Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[2] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proc. of SIGCOMM*, 2004.

[3] M. Balazinska and P. Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. In *Proc. of MobiSys*, 2003.

[4] Broadcom WLAN Chipset for 802.11a/b/g. `www.hotchips.org/archives/hc15/2_Mon/11.broadcom.pdf`.

[5] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement Driven Deployment of a Two-tier Urban Mesh Access Network. In *Proc. of Mobisys*, 2006.

[6] J. Camp, J. Robinson, C. Steger, and E. Knightly. Measurement Driven Deployment of a Two-tier Urban Mesh Access Network. In *Proc. of Mobisys*, 2006.

[7] Enabling Fast Wireless Networks with OFDM. `http://www.commsdesign.com/story/OEG20010122S0078`.

[8] JPL's Wireless Communication Reference Website. `http://wireless.per.nl/reference/chaptr03/fading/delayspr.htm`.

[9] G. Judd and P. Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *Proc. of NSDI*, 2005.

[10] K. Govil, E. Chan, and H. Wasserman. Comparing Algorithms for Dynamic Speed-Setting of a Low-Power CPU. In *ACM MobiCom 1995*, August 1995.

[11] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proc. of INFOCOM*, 2003.

[12] G. Holland, N. Vaidya, and P. Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. of MOBICOM*, 2001.

[13] P. Horowitz and W. Hill. *The Art of Electronics*. Cambridge University Press, New York, NY, USA, 1989.

[14] J. Proakis, Digital Communications, McGraw Hill, 2001.

[15] A. Kamerman and L. Monteban. WaveLAN-II: a high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, 2(3), 1997.

[16] D. Kotz and K. Essien. Analysis of a Campus-wide Wireless Network. *Wireless Networks*, 2005.

[17] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *Proc. of MSWiM*, 2004.

[18] M. Gast. 802.11 Wireless Networks: The Definitive Guide, Second Edition. O'Reilly & Associates, Inc., California, 2002.

[19] Clock Solutions for WiFi (IEEE 802.11). `http://www.pericom.com/pdf/applications/AN070.pdf`.

[20] E. Shih, P. Bahl, and M. Sinclair. Wake on Wireless: An event driven power saving strategy for battery operated devices. In *Proc. of MOBICOM*, 2002.

[21] D. Tang and M. Baker. Analyis of a Local-Area Wireless Network. In *Proc. of MOBICOM*, 2000.

[22] The Tolly Group. High Speed Wireless LANs: The Impact of Super G Proprietary Performance Mode on 802.11g Devices. `http://www.54g.org/`.

[23] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for Reduced CPU Energy. In *Proc. of OSDI*, 1994.

[24] Wimax forum whitepapers: http://www.wimaxforum.org/.

[25] I. Wormsbecker and C. Williamson. On Channel Selection Strategies for Multi-Channel MAC Protocols in Wireless Ad Hoc Networks. *Proc. of WiMob*, 2006.

[26] Y. Yuan, P. Bahl, R. Chandra, T. Moscibroda, and Y. Wu. Allocating Dynamic Time-Spectrum Blocks in Cognitive Radio Networks. In *Proc. of MOBIHOC*, 2007.