FLUID: Improving Throughputs in Enterprise Wireless LANs through Flexible Channelization

Shravan Rayanchu, Vivek Shrivastava, Suman Banerjee, and Ranveer Chandra

Abstract—This paper introduces models and a system for designing 802.11 wireless LANs (WLANs) using *flexible channelization*—the choice of an appropriate channel width and center frequency for each transmission. In contrast to current 802.11 systems that use fixed width channels, the proposed system, FLUID, configures all access points and their clients using *flexible channels*. We show that a key challenge in designing such a system stems from managing the effects of interference due to multiple transmitters employing variable channel widths, in a network-wide setting. We implemented FLUID in an enterprise-like setup using a 50 node testbed (with off-the shelf wireless cards) and we show that FLUID improves the average throughput by 59 percent across all PHY rates, compared to existing fixed-width approaches.

Index Terms—Channel width, conflict graph, scheduling, spectrum, WiFi.

1 Introduction

Traditionally, wireless channels strictly correspond to a predefined center frequency and a specific channel width. While this strict notion of a channel has served us well over the years, researchers in recent years have realized that flexible channels—channels in which the center frequency and bandwidth are picked based on traffic demands, noise and interference levels across a spectral band—can be particularly useful to improve spectrum efficiency. In the context of dynamic spectrum access networks and cognitive wireless networks, a large body of work [7], [15], [17], [25], [27], [28] has examined strategies to assign flexible channels. More recently, this problem of choosing the right frequency and width for communication has gained relevance with the onset of white-space networking where agile adaptation of these parameters is essential [6].

There has also been a growing attempt to explore the usefulness of flexible channels in the context of 802.11-based networks. Current 802.11 hardware can provide a limited amount of software-level flexibility that allows transceivers to operate on such flexible channels, e.g., a fixed number of channel widths (5, 10, 20, and 40 MHz) and a set of permissible center frequencies in the 2.4 or 5 GHz band [11]. Using this flexibility, the work in [8] shows how a single 802.11 link can pick an efficient channel width to adequately meet its traffic demand. At a high level, [8] shows that increasing channel width for a *single, isolated* link potentially allows greater throughput. But that, for a given total transmit power used by a wireless card, the power per unit frequency reduces for larger widths [8], leading to reduced SNR and poor connectivity in longer links.

Focus of FLUID. While the work in [8] focused on how to adapt the channel width for a *single, isolated link*, we focus on how to employ flexible channelization when using *multiple, potentially interfering links*. We look at the use of flexible channelization in a fairly complex and realistic setting—assigning flexible channels and improving throughput for an 802.11 enterprise WLAN using off the shelf hardware. The core problem we address in this paper is the following:

"Given an enterprise WLAN with many different Access Points (APs) and arbitrarily located wireless clients, how should flexible channels for each AP be structured?"

Initially, we imagined that the problem has an easy solution: identify the traffic demand for each AP (aggregated over all its clients) and provide a single channel to each AP that is proportional to this traffic demand. The channel choices can be periodically adapted based on demand evolution. Indeed, work in [22] proposes and shows the benefits of such a solution through careful simulation based studies. However, in our attempt to implement such a solution on an 802.11 testbed, we quickly uncovered new challenges.

One of the biggest challenges was to create an effective model for a *conflict graph*—a graph that captures the interference between a link and a potential interferer. Prior work (e.g., [22], [27], [28]) assumes that the interference behavior of two, potentially conflicting, links is unaffected by changes in their channel widths. However, in reality, the interference properties of two links can be greatly impacted by their channel width of operation, even if they use the same channel configuration (i.e., the same width and center frequency).

We illustrate this through a simple, yet interesting example. Given two links and a spectrum band, say 40 MHz, there are many ways to assign flexible channels (Fig. 1). Some natural choices are: 1) both links operate using the entire 40 MHz channel and time share using regular

- *S. Rayanchu, V. Shrivastava, and S. Banerjee are with the Department of Computer Sciences, University of Wisconsin, 1210 West Dayton St., Madison, WI 53706. E-mail: {shravan, viveks, suman}@cs.wisc.edu.*
- *R. Chandra is with the Mobile Computing Research Center, Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: ranveer@microsoft.com.*

Manuscript received 15 Nov. 2011; revised 26 Mar. 2012; accepted 29 Mar. 2012; published online 5 Apr. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org and reference IEEECS Log Number TMCSI-2011-11-0614.
Digital Object Identifier no. 10.1109/TMC.2012.89.

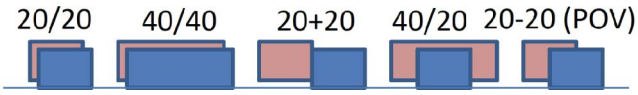

Fig. 1. Example flexible channel configurations using two channel widths of 20 and 40 MHz. The total available spectrum is 40 MHz.

random access mechanisms (**40/40** in Fig. 1), and 2) both links operate on separate 20 MHz channels (**20+20**) and potentially suffer no interference from each other. Initially, we assumed that examining these two choices alone is adequate to find the most efficient channel assignment. However, in our testbed experiments we found multiple two-link conflict scenarios where the best channel configurations were fairly nonstandard, including: 1) one link on a 40 MHz channel, the other on a 20 MHz channel, both with the same center frequency (**40/20**), 2) both links on partially overlapped 20 MHz channels, **20-20(POV)**. Interestingly, we also found several cases where using a single 20 MHz channel (**20/20**) provided better throughput than operating the links on a single 40 MHz channel (40/40).

The reason these other channel choices proved to be the best configuration for some link topologies was due to the variable nature of conflict that changes with channel width, even when the center frequency of the two links is identical. In fact, through experiments we found that changing channel widths has a great impact on all wireless interference parameters, e.g., carrier sense and interference range, hidden terminals, exposed terminals, etc. There were many instances where two neighboring links were in carrier sense range when using the same 20 MHz, but turned into hidden terminals when their channel widths were identically increased to 40 MHz. Exposed terminal scenarios sometimes appeared when reducing channel widths. More complex interference patterns arose in the presence of multiple links, and when considering different center frequencies, since some of the assignments resulted in partial spectral overlaps.

Hence, in our overall problem of assigning flexible channels in an enterprise WLAN, we have to compute the conflict graph *for all possible channel widths and center frequencies*. For an N node network using $|w|$ possible channel widths and k PHY data rates (e.g., for 802.11a, $k = 8$), this can require $O(N^2 \cdot k \cdot |w| \cdot 2^{|w+1|})$ measurements, one for each link pair, data rate, channel width, and center frequency (Section 4). This is a particularly daunting and complex task. To address this, we develop techniques to model the conflict graph using only $O(N \cdot k)$ empirical measurements at a *single* channel width. The next step is to use this conflict graph to assign flexible channels. In our proposed system, FLUID, a central controller improves the network throughput by assigning the center frequencies and widths to the APs on the fly, depending on the actual traffic demand. To further maximize the number of simultaneous transmissions, FLUID explores a joint data scheduling and flexible channelization approach. As we show in Section 5, the search space in this context grows exponentially in the number of transmissions. To tackle this, we propose a randomized algorithm with relatively low overhead to derive efficient transmission schedules, as demonstrated in our experiments.

We implemented FLUID on Atheros wireless cards running the MadWiFi driver [2] and have deployed the system on a 50 node testbed spanning multiple floors in our university building. Testbed results show that FLUID improves the median throughput by 59 percent across all possible PHY rates and when using dynamic rate adaptation, in a network-wide setting, compared to an approach using fixed width channels. To the best of our knowledge, FLUID is the first realization of an 802.11-based WLAN system consisting of multiple APs that are capable of operating at variable channel widths.

1.1 Key Contributions

Our contributions are as follows:

- We show that while flexible channelization can improve system throughput, its benefits in a network-wide setting are not immediate—careful construction of flexible channels requires taking into account the interference parameters like carrier sensing, hidden terminals, etc., which *depend* on the combinations of frequencies and channel widths used, as well as the specifics of topology and traffic demand (Section 2).
- We develop a modeling framework to efficiently compute the conflict graph for an N node network employing flexible channelization using only $O(N.k)$ empirical measurements at a *single* channel width, as opposed to brute-force approaches, which require $O(N^2.k.|w|.2^{|w+1|})$ measurements (Section 4).
- We present an algorithm to construct flexible channels, and show that combining flexible channelization with data scheduling can further improve network throughput (Section 5).
- Through a real deployment on our testbed, we evaluate FLUID over a variety of scenarios, and show that it can significantly improve the performance of a WLAN (Section 7).

2 Properties of Flexible Channels

Prior experimental work has noted three properties of varying channel widths on a *single, isolated* link [8]: 1) throughput of a link is proportional to the channel width, 2) halving the channel width doubles the power per Hertz, and consequently increases the range by 3 dB,[1] and 3) reducing the width by reducing the clock rate (and hence subcarrier spacing) results in lower battery consumption. One would expect the first two properties, in particular, link throughput, to be impacted by the interference from the other links in the network. In the rest of this section, we show that this is indeed the case and investigate the reasons behind this. Additionally, we show why designing a network that uses flexible channelization presents new challenges.

Measurement methodology. We perform measurements on a 50 node testbed deployed across five floors of a building. Each node runs Linux 2.6.20 kernel and is equipped with two Atheros 5212-based 802.11 NICs. Modifications to the MadWiFi driver allowed us to write to the hardware register that configures the PLL, giving us the capability to use four channel widths of 5, 10, 20, and

1. In Section 8, we discuss how our models can be modified to work in systems where this property might not hold.

TABLE 1
Choosing the Right Width Is Nontrivial as Throughput may not be Proportional to Channel Width under Interference

	20 ⟶ 10 MHz % links w/ Norm. Thr.			20 ⟶ 40 MHz % links w/ Norm. Thr.		
PHY Rate	0.5×	0.5×—1×	≥ 1×	2×	1×—2×	< 1×
Fixed 6 Mbps	44%	41%	15%	31%	45%	24%
Fixed 12 Mbps	42%	45%	13%	29%	48%	23%
Fixed 36 Mbps	37%	44%	19%	24%	49%	27%
Fixed 54 Mbps	38%	41%	21%	20%	51%	29%
SampleRate	38%	39%	23%	27%	45%	28%

Plot shows UDP throughputs for 10 and 40 MHz widths (throughputs normalized w.r.t. 20 MHz) across 2,872 link/interferer combinations for different fixed PHY rates and for dynamic rate adaptation (SampleRate). Shaded portion indicates the percentage of links for which the throughput is doubled (halved) when the width is doubled (halved).

40 MHz. We also made modifications to 802.11 timing parameters to ensure fair contention among different widths [8]. Experiments were carried out using 802.11a to avoid any external interference from our department WLAN that operates on 802.11 b/g. We experimented with *dynamic rate adaptation* and with *all fixed PHY data rates*, i.e., 6 to 54 Mbps in the 802.11a system. Due to space constraints, we typically present a snapshot of results, often using three fixed PHY rate scenarios (12, 36, and 54 Mbps[2]), as well as when the SampleRate algorithm [2] is used to dynamically adapt the PHY rate across all possible 802.11a rates. For bandwidth tests, the nodes broadcast 1,400 byte UDP packets at full sending rate for 10 seconds and experiments are repeated for 30 runs.

2.1 Impact of Flexible Channels

We observed that, in isolation, the throughput for high SNR links nearly doubles on doubling the channel width. However, in the presence of even one interferer, this property no longer holds. To show this, we randomly picked a 40 MHz interferer, and measured how the throughput of a randomly chosen good quality link (delivery ratio > 0.99) changes when it switches from 20 MHz to any of the other widths. The interferer and the link used the same center frequency. Table 1 shows the throughputs obtained at 40 and 10 MHz (throughput normalized w.r.t. 20 MHz) for four different PHY rates and rate adaptation (SampleRate), across 2,872 link/interferer combinations. Since the transmitter might be outside the interference range of a link, we observe that throughput doubles on doubling width for a certain fraction of the links, e.g., when using rate adaptation, 27 percent of the links doubled the throughput when switched from 20 to 40 MHz. However, this property doesn't hold in most other cases (unshaded portions in the table). For 45 percent of the links, the throughput *increases by varying amounts*—1× to 2×, and for the remaining 28 percent of the links, the throughput *decreases* when switched to 40 MHz. This holds for other widths as well, even though at varying degrees. In order to isolate the effect of PHY rate, we repeated the experiments across different fixed PHY rates and observed similar results (Table 1). To study why this happens, we look at the impact of widths on: carrier sense range, hidden and exposed links.

2. The data rate notations used in the paper correspond to the PHY rates when the channel width is set to 20 MHz (the default in 802.11). For example, 6 Mbps refers to OFDM with BPSK and coding rate of 1/2. The actual data rate would be doubled (or halved) when the channel width is set to 40 (or 10) MHz.

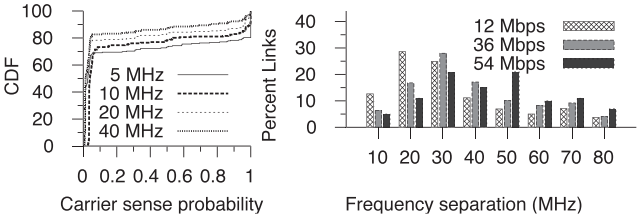

Fig. 2. (left) Carrier sensing probability at different widths for 600 link pairs. (right) Frequency separation needed for conflicting 40 MHz links to become nonconflicting at different PHY rates.

Carrier sensing range. Since smaller widths have higher energy per Hertz [8], we observed that more links carrier sense each other at lower widths. Fig. 2 (left) presents the CDF of carrier sensing probabilities among 600 link pairs in our testbed for different widths. Around 33 percent of link pairs carrier sense each other at 5 MHz, while only 15 percent of link pairs carrier sense each other at 40 MHz.

Hidden and exposed links. Table 2 shows the number of hidden and exposed links at different channel widths (and rates). While we find that the number of hidden and exposed links *vary with widths*, there is no particular trend. This is because lower widths not only cause more links to be in carrier sensing range, but also interfere over longer distances.

Partial spectrum overlaps. The extent of interference between links also depends on the amount of spectral overlap [19]. In case of flexible channels, partial spectral overlaps can occur when links use same center frequencies but different channel widths, or if links operate at different center frequencies. Such an interaction between the links has to be well understood in order to assign channels efficiently. For example, we observed that varying amount of frequency separation is needed between two conflicting 40 MHz links to make them noninterfering. Fig. 2 (right) shows this for 279 link pairs when using different PHY rates. At 36 Mbps, only 17 percent of the link pairs require a separation of 40 MHz. Fifty one percent require less separation (offering the opportunity for spectrum reuse). The remaining 32 percent require more than 40 MHz of separation, implying that naively packing these links at a separation of 40 MHz can degrade throughput.

2.2 Constructing Flexible Channels

We now study the impact of the above properties when assigning spectrum to links in a network. To begin with, we ask a simple question: *"If a total of 40 MHz of spectrum is available, how should we assign it to two links?,"* and show that the solution has many interesting considerations. The best

TABLE 2
Number of Hidden and Exposed Links Depend on the Channel Widths

PHY Rate	Scenario	5 MHz	10 MHz	20 MHz	40 MHz
Fixed 12 Mbps	Hidden	78	127	81	46
	Exposed	139	114	117	149
Fixed 36 Mbps	Hidden	81	135	87	58
	Exposed	121	108	84	96
Fixed 54 Mbps	Hidden	96	141	97	74
	Exposed	107	99	73	69

The precise methodology to identify hidden and exposed links was taken from [26].

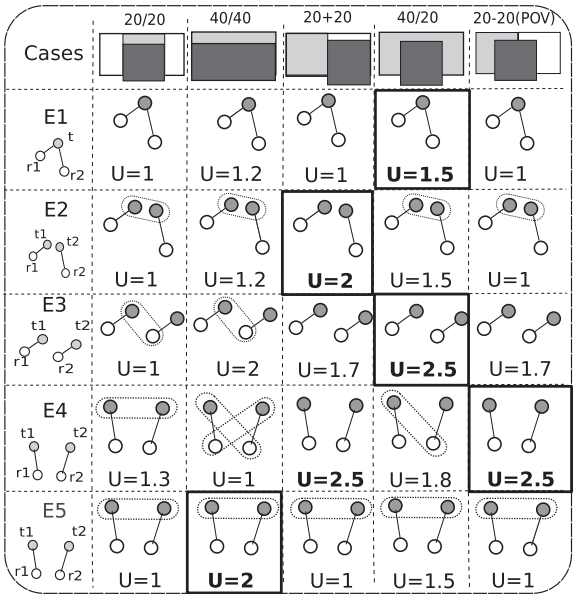

Fig. 3. Conflict information and corresponding throughputs with different spectrum assignments for real topologies in our testbed. A rounded rectangle enclosing two nodes represents a conflict (i.e., carrier sensing when the nodes are both transmitting, and interference when one is transmitting and the other node is receiving).

frequency and width assignment, changes depending on the topology and the interference among links.

Fig. 3 shows throughput measurements for five simple two-link topologies taken from real instances in our testbed along with the five example spectrum assignments described in Section 1. Here, the configuration 40/20 refers to the link (t_1-r_1) operating on the entire 40 MHz and (t_2-r_2) operating on 20 MHz, using the same center frequency. The configuration 20-20(POV) refers to the partial overlap case where the two links use two 20 MHz channels with center frequencies separated by 10 MHz. A rounded rectangle enclosing two nodes represents a conflict (i.e., carrier sensing when the nodes are both transmitting, and interference when one node is transmitting and the other is receiving). The first column shows the topology information, while the rest of the columns illustrate how the *conflicts change* across different assignments and result in different throughputs. The throughput values (U) are normalized w.r.t. to the lowest throughput for each case. For the cases that require channel/width switching (case E1, $20 + 20$, 40/20, and 20-20 (POV)) we use optimizations to reduce the switching overhead (Section 6). In all measurements, the traffic was backlogged on both links. For ease of exposition, in this section we present the results when the 802.11 PHY was set to the base rate of 6 Mbps.

We now briefly explain why the best spectrum assignment (shown in bold squares) differs in each case. **Case E1** in Fig. 3 corresponds to the scenario where client r_2 has a low SNR and thus a poor delivery ratio at 40 MHz; the delivery ratio increases to 1 at 20 MHz because of 3 dB increase in SNR. For client r_1, the delivery ratio is 1 at both widths. Here, using *client-centric* widths (40/20 in Fig. 3) achieves the best throughput (a gain of 25 percent over 40/40). All other configurations have worse throughputs as they either waste spectrum or result in a poor delivery ratio for r_2.

We consider two links in **Case E2**, with link (t_2-r_2) having a poor delivery ratio at 40 MHz. Using 40/20 improves the delivery for r_2 (due to increase in SNR of the link) compared to 40/40. However, the links still continue to carrier sense because of which they cannot effectively use the entire 40 MHz spectrum. Here, $20 + 20$ achieves a better throughput (a gain of 33 percent over 40/20) as both links can simultaneously operate on separate 20 MHz channels with good delivery ratios.

Case E3 illustrates the scenario of a one-way hidden terminal (t_1 interferes with r_2) which is resolved by separating the links on two 20 MHz channels ($20 + 20$). However, simply *narrowing the width resolves the conflict*—operating the link (t_2-r_2) at 20 MHz improves the SINR and hence makes the links nonconflicting. 40/20 improves the throughput by 47 percent over $20 + 20$ due to increased transmission concurrency.

In a two-way hidden terminal scenario (**Case E4**), the best configurations resolve the conflict between two links, either $20 + 20$, or partially overlapping assignment, 20-20(POV). Using 20-20(POV) might be more preferable for larger network scenarios as it uses lesser spectrum. Interestingly, using a single 20 MHz channel for both links (20/20) provides a better throughput than using a single 40 MHz channel (40/40), as the links carrier sense each other in the 20/20 configuration due to increase in their signal strengths.

Finally, **Case E5** represents a scenario where the links carrier sense each other when using the 20/20 configuration. However, it turns out that the links continue to carrier sense at all other configurations. For example, a center frequency separation of 20 MHz ($20 + 20$) is not adequate to resolve the carrier sensing conflict. Given this scenario, sharing the medium using 40/40 turns out to be the best configuration.

Why is the best flexible channel configuration different in each case? While the above is by no means an exhaustive set of flexible channel configurations, even exploring this limited set of configurations allowed us to observe cases where different configurations performed the best. This is because conflicts (carrier sensing and interference relationships) between the links are determined by the flexible channel configuration used. These conflicts in turn determine the total throughput in each case. Thus, in each case we have to choose a configuration that minimizes the conflict and improves the overall throughput. Put another way, to maximize the overall network throughput, we have to employ a *conflict-aware* mechanism which intelligently chooses a particular flexible channel configuration based on the carrier sensing and interference relationships at different widths and center frequencies.

3 FLUID: Overview

We propose FLUID, a system that improves the wireless network throughput through the use of flexible channels. While the design of FLUID is generic and can be applied to any 802.11-based setting, in this work, we focus on its application to an enterprise WLAN setting.

Target Network Setting. Consider an enterprise WLAN setting where clients and APs are capable of operating on flexible channels. All the APs are connected over an Ethernet backplane, and are managed using a central

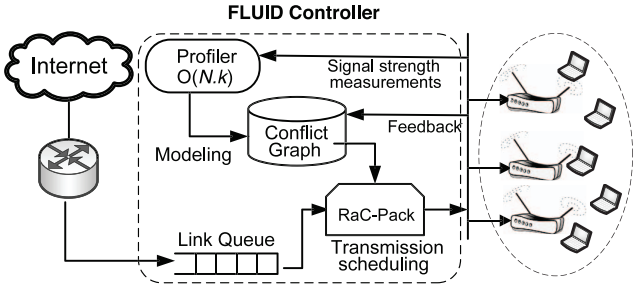

Fig. 4. Flow of operations in FLUID. Periodic signal strength measurements are used to update the modeled conflict graph (Section 4). Packets arrive from the network gateway and are enqueued at a central controller. The controller releases these packets based on the transmission schedules derived by a packing algorithm (Section 5). APs receive the packets and transmit them according to the controller's prescribed flexible channel assignment, and subsequently notify the controller of all failures. The controller uses this feedback for scheduling retransmissions and refining the conflict graph.

controller. Let B be the total amount of spectrum in use. Let $|w|$ denote the total number of channel widths to choose from. Let w_{min} denote the minimum channel width used, and assume that channel widths are of the form $w = w_{min} \cdot 2^r$, where $0 \leq r \leq |w| - 1$. In our implementation, $|w| = 4$ and $w_{min} = 5$, as we use 5, 10, 20, and 40 MHz as the possible channel widths. Let $\mathcal{F} = \{(f_c, w)\}$ be the set of permissible center frequency and width combinations s.t. f_c is of the form $f_c = w_{min} \cdot c$, where c is an integer and $[f_c - \frac{w}{2}, f_c + \frac{w}{2}] \subset [0, B]$.

We now sketch the main operations of FLUID. Fig. 4 illustrates the different components involved in FLUID.

1. Conflict Graph Generation. FLUID builds a conflict graph to model the interference between links while taking into account the combination of channel widths and center frequencies. Using a brute-force approach for conflict graph computation becomes infeasible as it requires $O(N^2 \cdot k \cdot |w| \cdot 2^{|w+1|})$ measurements. As discussed in Section 4, FLUID uses modeling techniques to reduce the overhead to $O(N \cdot k)$.

2. Interference Mitigation. The controller uses the conflict graph to mitigate interference and improve system throughput either by employing 1) an unscheduled approach, i.e., flexible channelization with DCF or 2) flexible channelization along with a scheduled approach such as CENTAUR [26], which can improve downlink performance. While we have explored both the approaches, in this paper, we focus on the harder problem of improving downlink system throughput using a *joint scheduling and flexible channelization* approach. Although designing such a scheduled system is more challenging than its unscheduled counterpart, it offers better performance than DCF with static channel assignment mechanisms for the following reasons: 1) it uses spectrum efficiently as it takes the actual traffic into consideration, 2) it resolves downlink hidden interference and opportunistically capitalizes on exposed terminal scenarios, 3) using a scheduled approach enables an AP in FLUID to employ *client-centric* widths which is otherwise difficult to manage with DCF in the presence of upload traffic. In Section 7, we show that FLUID's scheduled approach performs better than CENTAUR and the unscheduled approaches across various scenarios.

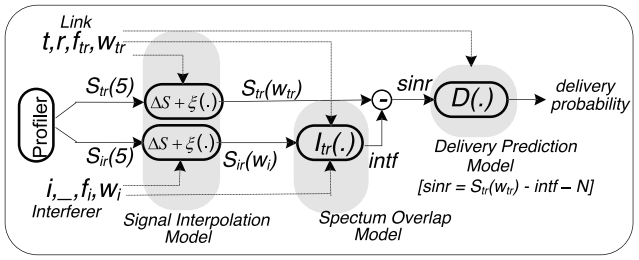

Fig. 5. Sketch of the modeling process. Signal strengths of the transmitter and the interferer at their respective widths are interpolated using their corresponding signal strengths at 5 MHz. The amount of interference is then computed based on the spectral overlap, which is used to calculate the SINR. Finally, the SINR is input to the delivery prediction model to compute the delivery under interference.

4 Modeling Conflicts in FLUID

In a traditional WLAN that uses a fixed channel width, the conflict graph between N transmissions (all on the same channel) can be generated by performing pairwise link throughput tests [12] at each PHY rate k, which requires a total of $O(N^2 \cdot k)$ measurements. Recent research [4], [14] has shown that this overhead can be reduced to $O(N \cdot k)$ using SINR-based modeling. Applying such models to a variable channel width system is not straightforward, as the number of spectral overlaps (and hence interference) depends on the combinations of center frequencies and channel widths used. Fig. 8 shows two example spectrum overlap configurations. The number of distinct nonzero spectrum overlap configurations using the set of permissible center frequencies (as detailed in Section 3) for two links operating on channel widths w_1 and w_2 can be calculated as $(w_1 + w_2)/w_{min} - 1$. Hence, the total number of spectrum overlap configurations taking into account $|w|$ possible widths are $\sum_{w_1} \sum_{w_2} ((w_1 + w_2)/w_{min} - 1)$, which evaluates to $2 \cdot |w| \cdot (2^{|w|} - 1) - |w|^2$. Thus, computing the conflict graph using the approach in [12] would now require a significant overhead of $O(N^2 \cdot k \cdot |w| \cdot 2^{|w+1|})$, making it intractable for real systems. Next, we show how our models significantly reduce this measurement overhead.

Modeling overview. The goal of the conflict graph module in Fig. 4 is to predict the delivery ratio on a link (transmitter-receiver pair) in the presence of an interferer. It uses SINR-based empirical models to predict the delivery probabilities. In what follows, we first explain how our model computes the SINR for *perfect spectral overlap* case (the link and the interferer use the *same* center frequency and width), at all channel widths, using only measurements at a single width. We then extend the model to compute the SINR for *partial spectral overlap* case (the link and the interferer can use different center frequencies and widths). Finally, we derive the delivery prediction models using empirical measurements and use the computed SINR to model the delivery under interference. Fig. 5 shows the overall modeling process. The parameters used in the modeling procedure are summarized in Table 3.

Interpolating SNR at different widths, using single width measurements. To compute the SINR at the receiver, we have to measure the signal strengths of the transmitter and the interferer at the receiver. However, as we show

TABLE 3
Parameters Used in FLUID's Modeling Procedure

Modeling parameter	Definition
P_i	Transmitted power per unit Hz (at width w_i)
$\mathcal{A}(.)$	Signal attenuation function
$\widehat{S}_{tr}(w_i)$	Signal strength (in dBm) per hertz between transmitter t and receiver r using width w_i
$\Delta S(w_i, w_j)$	Modeled difference in received signal strength per hertz when switching from width w_i to width w_j
$\xi(w_i, w_j)$	Correction function applied to $\Delta S(w_i, w_j)$ to improve the model accuracy
$\mathcal{N}$	Noise floor per hertz
$\mathcal{I}_{t,r}(\tau, w_t, w_r)$	Quantifies the spectral overlap between a transmitter t using center frequency and width (f_t, w_t) and a receiver r using center frequency and width (f_r, w_r). Here $\tau = \lvert f_t - f_r \rvert$
$B_{r,w_r}(f)$	Band-pass filter's frequency response when a channel width of w_r MHz is used
$\widehat{\mathcal{D}}(.)$	Function used to predict delivery ratio based on modeled SINR

below, the received signal strength per hertz depends on the channel width. This would require us to carry out signal strength measurements at every channel width, resulting in a measurement overhead of $O(N{\cdot}|w|)$. We now show that it is possible to interpolate the received signal strength per hertz at different widths from measurements at only one width.

Let P_i and P_j be the transmitted power per unit Hz at widths w_i and w_j, respectively. Since the total power transmitted by the card is the same in both cases, we have $P_i{\cdot}w_i = P_j{\cdot}w_j$. Now, the signal strength per hertz at the receiver depends on the attenuation experienced by the wireless signal and is given by $s_i = \mathcal{A}(P_i)$. We can approximate the attenuation $\mathcal{A}(.)$ as $d^{-\alpha}P_i$, where α is the path-loss exponent [10]. We can compute the difference in received signal strength per hertz, $\Delta S(w_i, w_j)$ as $10log(\frac{s_i}{s_j}) = 10log(\frac{P_i}{P_j}) = 10log(\frac{w_j}{w_i})$.

However, we observed that the difference in signal strength per hertz for our hardware only follow this relationship approximately. When we decreased the channel width from 40 to 5 MHz, we observed $\Delta S(w_i, w_j)$ to be 8.6 dB on average, instead of 9 dB (per unit Hz). To account for this difference, we introduce a correction function $\xi(.)$. Let $\widehat{S}_{tr}(w_i)$ denote the signal strength per hertz (in dBm) between transmitter t and receiver r at width w_i, derived using empirical measurements. We have

$$\widehat{S}_{tr}(w_i) = \widehat{S}_{tr}(w_j) + \Delta S(w_i, w_j) + \xi(w_i, w_j). \quad (1)$$

We empirically calculate the value of $\xi(.)$ using signal strength measurements from our testbed. We assume the noise floor per hertz ($\mathcal{N}$) to be constant and the signal to be evenly distributed over the transmitted bandwidth. We calculate the SNR at width w as $\widehat{S}_{tr}(w) - \mathcal{N}$. Fig. 9 (left) shows the CDF of signal strengths at different widths for all links in our testbed. We observed that the difference between measured and theoretical signal strength per hertz values does not vary significantly, even for the most bursty link in our testbed. The observed mean/std. deviation values *across all links* for $\xi(40, 5)$ were $-0.34/0.13$ dB, that for $\xi(20, 5)$ were $-0.13/0.12$ dB, and finally for $\xi(10, 5)$ were $-0.08/0.16$ dB (per unit Hz). Since these variations are low, in our model, we account for the difference in the measured and theoretical signal strength per hertz using the mean value of $\xi(.)$. We note that this calibration is a one-time overhead that is necessitated when using the model for the first time with a particular hardware chipset. Once the mean values of $\xi(.)$ are determined, these can be used to accurately and efficiently estimate SNR as follows: Instead of carrying out the signal measurements at every width, we carry out $O(N)$ signal measurements at the lowest width of 5 MHz (as it has the longest range), and use (1) to derive the SNR at all other widths.

Modeling SINR for perfect spectral overlaps at all widths. To model SINR in the presence of an interferer, using width w, we first interpolate the signal strength per hertz of the transmitter to the receiver, and that of the interferer to the receiver, i.e., we use (1) to interpolate $\widehat{S}_{tr}(w)$ and $\widehat{S}_{ir}(w)$ from corresponding signal measurements at 5 MHz, $\widehat{S}_{tr}(5)$ and $\widehat{S}_{ir}(5)$. Now, the SINR can simply be calculated as $\widehat{S}_{tr}(w) - \widehat{S}_{ir}(w) - \mathcal{N}$ dB. We now provide extensions to the previous model, to quantify the amount of interference for the partial overlap case where the links can use any permissible center frequencies and channel widths.

Modeling SINR for partial spectral overlaps at all widths and frequencies. To characterize the amount of interference experienced by a receiver r using a width w_r and a center frequency f_r, from an interferer t using a width w_t and center frequency f_t, we extend the model developed in [19] to calculate the *interference factor*, $\mathcal{I}_{t,r}(.)$ for a variable channel width system. $\mathcal{I}_{t,r}(.)$ quantitatively captures the amount of spectral overlap between the interferer and the receiver by calculating the area of intersection between a signal's spectrum and a receiver's band-pass filter. We incorporate the interferer and receiver channel bandwidths, w_t and w_r into this model to derive $\mathcal{I}_{t,r}(.)$:

$$\mathcal{I}_{t,r}(\tau, w_t, w_r) = \int_{-\infty}^{+\infty} T_{t,w_t}(f) B_{r,w_r}(f - \tau)\, df. \quad (2)$$

In above equation, the parameter τ represents the difference in the center frequencies of the channels, i.e., $\tau = f_t - f_r$. The parameter $T_{t,w_t}(f)$ denotes the transmitted signal's power distribution across the frequency spectrum when a channel bandwidth of w_t MHz is used. We approximate $T_{t,w_t}(f)$ with the corresponding transmit spectrum mask [19]. Finally, $B_{r,w_r}(f)$ denotes the band-pass filter's frequency response when a channel of w_r MHz is used. Assuming the receive filter for a particular bandwidth to be same as the transmit spectrum mask [19], for 802.11a we get

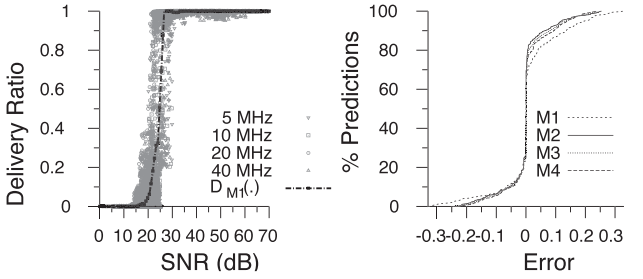

Fig. 6. (left) Delivery ratio as a function of mean signal strength for different widths, across all the receivers at 6 Mbps. We show measured delivery ratio values and piecewise linear interpolation as a function of SNR (model M1). (right) CDF of modeling error for all the four models.

$$B_{r,w_r}(f) = T_{t,w_t}(f) = \begin{cases} -40 \text{ dB} & \text{if } |f - F_c| \geq (30/\mathcal{B})\text{MHz} \\ -28 \text{ dB} & \text{if } (20/\mathcal{B})\text{MHz} \leq |f - F_c| < (30/\mathcal{B})\text{MHz} \\ -20 \text{ dB} & \text{if } (11/\mathcal{B})\text{MHz} \leq |f - F_c| < (20/\mathcal{B})\text{MHz} \\ 0 \text{ dB} & \text{otherwise,} \end{cases} \quad (3)$$

where F_c denotes the channel center frequency and bw is the channel bandwidth (w_t or w_r) used and $\mathcal{B}$ is the bandwidth scaling factor calculated as $\mathcal{B} = 20/bw$.

Now for two links (t_1, r_1) and (t_2, r_2) using center frequencies and widths (f_1, w_1) and (f_2, w_2), the amount of interference experienced by r_1 can be characterized as $intf = \widehat{S}_{t_2 r_1}(w_2) + 10log(\mathcal{I}_{t_2,r_1}(|f_2 - f_1|, w_2, w_1))$ dB. The effective SINR would be $\widehat{S}_{t_1 r_1}(w_1)$-$intf$-$\mathcal{N}$ dB.

Predicting delivery ratio. In the last step of our modeling process, we predict the delivery ratio for a link using the SINR estimated earlier. We first show the relationship between SNR and the delivery ratio for an isolated link when using different widths, and then derive delivery prediction models.

Delivery under isolation. We perform $O(N \cdot |w| \cdot k)$ measurements where each node broadcasts in turn at all widths and rate combinations, and the remaining nodes measure the average signal strengths and corresponding delivery ratios. All nodes use the same center frequency and channel width. Fig. 6 (left) shows the SNR versus delivery ratio for 231 link pairs for each of the four channel widths at 6 Mbps.[3] For values of SNR greater than 26 dB, the delivery ratio is close to 1, whereas for SNR less than 18 dB, the deliver ratio is close to 0; for intermediate values of SNR, the delivery ratio increases with signal strength. This behavior is *similar* across widths, since for a given signal strength, the probability that a packet is successfully decoded is independent of width. Furthermore, we observed a stronger correlation between SNR and delivery ratio when viewed across individual receivers.

Based on this, the most relevant parameters for modeling delivery are: SNR, channel width and the receiver under consideration. In light of this, we explored four models to derive the delivery prediction function $\widehat{\mathcal{D}}(.)$. In M1, we model the delivery ratio as a piecewise linear function of SNR. In M2, we used receiver-specific curves including the SNR and channel width. M3 only used receiver-specific curves along with SNR. M4 is similar to M3, except that SNR is computed using (1).

3. This behavior also holds for all the other rates. The SNR curves are shifted to the right, as higher rates require a higher SNR to decode a packet correctly.

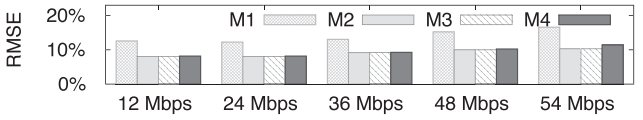

Fig. 7. Prediction error for all the models at different PHY rates.

Delivery under interference. To predict the delivery under interference, we compute the SINR using the techniques mentioned before and feed this into one of the four delivery prediction models. We now evaluate the accuracy of these models in the presence of an interferer for the perfect spectral overlap case.

In order to measure the ground truth, we carry out the following $O(N^2 \cdot k \cdot |w|)$ measurements: we pick a pair of nodes in turn, and both of them simultaneously transmit data while the rest of the nodes measure the signal strengths and corresponding delivery ratios. This process is repeated for all channel width and rate combinations. We note that all nodes use the same center frequencies and widths. Fig. 6 (right) shows the CDF of the error for all the four models at 6 Mbps, and Fig. 7 shows the root mean square error (RMSE) for the models across different PHY rates. We observe that all the four models perform reasonably well. Models M2, M3, and M4 have lower error compared to M1, owing to the use of receiver specific curves. For these models, the error is less than 10 percent for 90 percent of the predictions, with maximum error being less than 30 percent (Fig. 6 (right)). The overall RMSE for all the models were: 14.2, 8.7, 8.9, and 9.6 percent. We observe that M2 and M3 have very similar performance, confirming that the delivery ratios were independent of the width used. More importantly, M4 which uses signal interpolation has an accuracy which is quite close to M2. This is a useful result as it helps us reduce the conflict graph computation overhead to $O(N \cdot k)$ for a network where all links can operate on any width while using the same center frequency. We therefore choose M2 for delivery prediction in FLUID. We also evaluated the models for the partial overlap case, and observed similar delivery prediction accuracy numbers.

Packing accuracy. We now evaluate both the partial and perfect spectrum overlap cases using a more intuitive measure—error in predicting the minimum frequency separation required to resolve the conflict between any two links. Note that overpredicting the frequency separation leads to poor usage of spectrum, while underprediction can result in throughput degradation.

We experimented with 500 link-interferer (tr-i) combinations (across different PHY rates) in our testbed, where the link and the interferer can use any widths, w_{tr} and w_i. In each case we measured $\widehat{f}_{min}$, the minimum frequency separation required between the link and the interferer such that the conflict is resolved. We also compute the predicted separation f_{min} using the $\mathcal{I}_{t,r}(.)$ model, and a naive packing approach where the center frequencies are simply separated by $(w_{tr} + w_i)/2$ MHz. We then compute the difference in the measured and predicted frequency separation $\Delta f_{min} = f_{min} - \widehat{f}_{min}$. Fig. 9 (right) shows the CDF of Δf_{min} for both the models. The $\mathcal{I}_{t,r}(.)$ model results in better spectrum reuse by predicting Δf_{min} correctly in

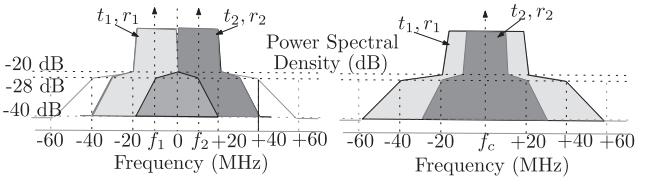

Fig. 8. Example spectrum overlap scenarios. (left) Two links (t_1, r_1) and (t_2, r_2) using a channel width of 20 MHz and center frequencies f_1 and f_2 separated by 20 MHz. (right) Two links (t_1, r_1) and (t_2, r_2) using the same center frequency (f_c), but different channel widths of 40 and 20 MHz, respectively.

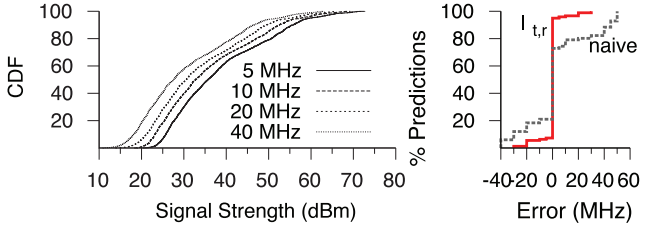

Fig. 9. (left) CDF of signal strengths in the testbed for different channel widths. (right) CDF of error in estimating the minimum channel separation (Δf_{min}) across different PHY rates and channel width combinations, using naive and $\mathcal{I}_{t,r}$ models.

87.6 percent of the cases. The naive model predicts only 52 percent of the cases accurately.

Example scenarios. We now present two example scenarios and show how $\mathcal{I}_{t,r}(.)$ can be used to model the interference.

Example 1. *How much frequency separation is needed?* Consider the packing in Fig. 8 (left), where the two 20 MHz links (t_1, r_1) and (t_2, r_2) have their center frequencies separated by of 20 MHz. Is the frequency separation enough? To answer this we have to estimate the delivery ratio at both the receivers for this packing: the interference factor for r_1 in this case turns out to be $\mathcal{I}_{t_2,r_1}(20, 20, 20) = 0.0571$, which is reduction in the interferer signal by 12.43 dB. Hence, the SINR at r_1 in this case would be $\widehat{S}_{t_1 r_1}(20) - \widehat{S}_{t_2 r_1}(20) + 12.43$ dB. If this is not sufficient for $\widehat{\mathcal{D}}(r_1, SINR)$ to be close to 1, then the links have be separated further apart. For example, at 25 MHz, the $\mathcal{I}_{t_2,r_1}(25, 20, 20) = 0.0081$ which results in SINR at r_1 of $\widehat{S}_{t_1 r_1}(20) - \widehat{S}_{t_2 r_1}(20) + 20.91$ dB, i.e., an increase in SINR by around 8.5 dB which might sufficient for the delivery ratio to be close to 1. Similarly, we can estimate the frequency separation needed for r_2.

Example 2. *Can we improve spatial reuse by narrowing widths?* Consider two 40 MHz links (t_1, r_1) and (t_2, r_2) which are on the same center frequency. The interference factor for both receivers is 1, SINR at r_1 is $\widehat{S}_{t_1 r_1}(40)$-$\widehat{S}_{t_2 r_1}(40)$ and that at r_2 is $\widehat{S}_{t_2 r_2}(40)$-$\widehat{S}_{t_1 r_2}(40)$. How does the interference relationship change when (t_2, r_2) switches to 20 MHz, while not changing the center frequency? The resulting packing is shown in Fig. 8 (right). When (t_2, r_2) switches to 20 MHz, the interference factor for r_2 remains the same (i.e., $\mathcal{I}_{t1,r2}(0, 40, 20) = 1$), since there is a complete spectral overlap for r_2. However, $\widehat{S}_{t_2 r_2}(20)$ is around 3 dB higher than $\widehat{S}_{t_2 r_2}(40)$ (1). This results in SINR at r_2 increasing by 3 dB. Whereas, for r_1, the interference factor $\mathcal{I}_{t2,r1}(0, 20, 40) = 0.49$, which is around -3.1 dB. Therefore, the new SINR becomes $\widehat{S}_{t_1 r_1}(40) - \widehat{S}_{t_2 r_1}(20) - 3.1$ dB, i.e., $\widehat{S}_{t_1 r_1}(40) - \widehat{S}_{t_2 r_1}(40) + 3 - 3.1$ dB implying, that the SINR at r_1 almost remains the same. That is, by narrowing the channel width of a link, we could improve its SINR without affecting the SINR of the other link. In our experiments, we found a number of such instances where narrowing channel width can provide opportunities for spatial reuse (Section 7).

External interference. In order to handle external interference, the model can be modified as follows: the receivers can measure the increased noise floor, and report this back along with the signal strength measurements. The SINR computation can then use a receiver specific noise floor $\mathcal{N}_r$, instead of a constant $\mathcal{N}$ to improve the accuracy of delivery prediction.

Summary. We sketch the modeling process in Fig. 5. We carry out $O(N \cdot k)$ measurements at the lowest channel width, 5 MHz. In order to predict the delivery ratio of link in the presence of an interferer, we first interpolate the signal strengths of the transmitter and the interferer at their widths. Based on the spectral overlaps, we compute the interference using the $\mathcal{I}_{t,r}(.)$ model. Finally, we calculate the SINR, which is then input to $\widehat{\mathcal{D}}(.)$ to estimate the delivery probability.

5 Transmission Packing

Assume that a set of packets arrive at the FLUID controller. Now, based on the conflict graph, the next step for the controller is to "pack" the transmissions, i.e., determine the subset of packets that can be scheduled for transmission simultaneously, along with an assignment of the center frequencies and channel widths. In FLUID, such a decision is made at the time granularity of an *epoch*. We discuss the factors that determine the epoch duration in Section 6.

Scheduling complexity. The scheduling problem to optimize throughput by assigning appropriate time-frequency blocks is NP-hard [28]. The size of this problem is $\sum_{r=1}^{r=N} \binom{N}{r} |\mathcal{F}|^r$, where $\binom{N}{r}$ is number of the ways in which the controller can pick r out of N transmissions, and $|\mathcal{F}|^r$ is all possible frequency and width combinations for r APs.

Packing heuristics. In order to reduce the search space in scheduling, we use two heuristics explained below:

Throughput estimation. The throughput estimation algorithm, *estimateTput()* (Algorithm 1) takes a set of packed transmissions $\mathcal{T} = (t, r, f, w)$, and returns a vector of estimated individual transmission throughputs. The throughput of an individual transmission T_i is calculated as follows: the effective signal strength from each of the other $\mathcal{T} - \{T_i\}$ transmissions is calculated using the modeling techniques presented in Section 4, and is summed up to calculate the total interference (lines 8-10). This then used to compute the SINR. Finally, the controller uses the SINR to estimate the throughput by picking the best PHY data rate (lines 14-19): it iterates through the delivery ratio curves for each data rate, and picks the rate which maximizes the throughput (data rate $\times$ delivery probability).

Algorithm 1: Model based Throughput Estimation (estimateTput)

Input : Set of packed transmissions $\mathcal{T} = \{t, r, f, w\}$, Delivery ratio model, $\widehat{\mathcal{D}}(.)$, Signal correction model, $\xi(.)$
Output: Estimated throughput for $\mathcal{T}$

1 $\overrightarrow{tv}_{tot} \leftarrow 0$
2 **foreach** $T_i = (t_i, r_i, f_i, w_i) \in \mathcal{T}$ **do**
3 $\quad (d_{T_i}^{\mathcal{T}-\{T_i\}}, dr_{best}) \leftarrow \rho(T_i, \mathcal{T} - \{T_i\})$
4 $\quad \overrightarrow{tv}_{tot}[i] \leftarrow (d_{T_i}^{\mathcal{T}-\{T_i\}} * dr_{best})$
5 **return** $\overrightarrow{tv}_{tot}$
6 **Procedure** $\rho(T_i, \mathcal{T}')$**:**
7 $\quad$ Let $T_i = \{t_i, r_i, f_i, w_i\}$; $intf \leftarrow 0$
8 $\quad$ **foreach** $T_j = (t_j, r_j, f_j, w_j) \in \mathcal{T}'$ **do**
9 $\quad\quad S_{t_j,r_i}(w_j) \leftarrow \widehat{S}_{t_j,r_i}(5) + \Delta\, S(w_j, 5) + \xi(w_j, 5)$
10 $\quad\quad S'_{t_j,r_i} = S_{t_j,r_i}(w_j) + 10\ log(\mathcal{I}_f(|f_j - f_i|, w_j, w_i))$
$\quad\quad intf = intf + S'_{t_j,r_i}$
11 $\quad S_{t_i,r_i}(w_i) \leftarrow \widehat{S}_{t_i,r_i}(5) + \Delta\, S(w_i, 5) + \xi(w_i, 5)$
12 $\quad sinr = S_{t_j,r_i}(w_i) - intf$
13 $\quad max\_tput \leftarrow 0$
$\quad$ /* SINR based rate adaptation */
14 $\quad$ **foreach** $dr \in R$ **do**
15 $\quad\quad cur\_tput \leftarrow dr * \widehat{\mathcal{D}}(r_i, dr, sinr)$
16 $\quad\quad$ **if** $cur\_tput > max\_tput$ **then**
17 $\quad\quad\quad max\_tput \leftarrow cur\_tput$
18 $\quad\quad\quad dr_{best} \leftarrow dr$
19 $\quad\quad\quad d_T^{\mathcal{T}'} \leftarrow \widehat{\mathcal{D}}(r_i, dr, sinr)$
20 $\quad$ **return** $(d_T^{\mathcal{T}'}, dr_{best})$

RaC-Pack. In FLUID, the central controller uses a randomized algorithm, RaC-Pack (Randomized Compaction based Packing) to derive the transmission schedules. RaC-Pack (Algorithm 1) takes the FIFO queue of packets at the controller as input and creates a set of packed transmissions for each epoch. We first describe the compaction step that can be applied to a packed transmission set so as to maximize a particular objective.

Compaction step. Keeping the center frequency and width assignments of all the other transmissions the same, the compaction step (lines 22-29) assigns a center frequency and width to a particular transmission, T_i that maximizes a criteria (lines 24-28). We supply the objective function (*computeOBJ*) with one of the following two criteria: 1) maximize the total throughput (FLUID-thr) or 2) find the best min-max throughput (FLUID-fair) which results in better fairness, at the cost of throughput. The function *estimateTput*, is used to estimate throughput during each iteration (line 26).

The RaC-Pack scheduling algorithm works as follows: In order to prevent starvation, RaC-Pack always schedules the first packet in FIFO queue for transmission in the current epoch. It then applies the compaction step to this transmission to find the "best" packing (lines 2-4). Next, the algorithm goes through the rest of the transmissions in a randomized order, and adds them to the transmission schedule if they improve the throughput (lines 5-20). This is done by adding a transmission to the currently packed set, and then repeatedly invoking the compaction step for the each of the transmissions in succession. The order of invocation is randomized by using a random permutation of the transmissions. This compaction process (lines 13-18) is repeated until the objective function stops improving. We note that this iterative process will converge, as in each iteration, the objective function progressively improves the throughput vector based on the specified criteria. The total number of rounds for the algorithm can vary with the topology and traffic pattern, and the worst case complexity is $O(|\mathcal{F}|^N)$. We set an upper bound of 50 rounds, and in our experiments with different topologies, we found that the algorithm converges after approximately 21.3 rounds on an average. In Section 7, we compare RaC-Pack to the brute-force approach of evaluating all possible schedules.

Algorithm 2: RaC-Pack: Transmission Packing

Input : fifoQ (FIFO queue of packets), $\mathsf{vQ}_1 \ldots \mathsf{vQ}_n$ (per-client virtual packet queues), $\mathcal{F} = \{(f, w)\}$ (set of frequency f, width w combinations)
Output: Set of packed transmissions $\mathcal{T}_{next} = \{(t, r, f, w)\}$

1 $\mathcal{T}_{next} \leftarrow 0$, $\mathcal{T}_{cur} \leftarrow 0$
2 $p_{head} \leftarrow$Dequeue(fifoQ); $(f_1, w_1) \leftarrow \mathcal{F}[0]$
3 $T_1 \leftarrow (\mathsf{tx}(p_{head}), \mathsf{rx}(p_{head}), f_1, w_1)$; $packedAPs \leftarrow \mathsf{tx}(p_{head})$
4 $(\mathcal{T}_{next}, \overrightarrow{tv}_{best}) \leftarrow$ COMPACTION$(\{T_1\}, 0)$; $\mathcal{T}_{cur} \leftarrow \mathcal{T}_{next}$
5 $r_i \leftarrow$ RAND$(0 \ldots n-1)$
6 **for** i *in* $0 \ldots n$ **do**
7 $\quad next \leftarrow (r_i + i)$ mod n
8 $\quad p_{next} \leftarrow$Dequeue(vQ $next$)
9 $\quad$ **if** tx $(p_{next}) \in packedAPs$ **then**
10 $\quad\quad$ continue
11 $\quad T_{next} \leftarrow (\mathsf{tx}(p_{next}), \mathsf{rx}(p_{next}), f_1, w_1)$
12 $\quad \mathcal{T}_{cur} \leftarrow \mathcal{T}_{next} \bigcup T_{next}$
13 $\quad$ **while** $\mathcal{T}_{cur} \neq \mathcal{T}_{prev}$ **do**
14 $\quad\quad \mathcal{T}_{prev} \leftarrow \mathcal{T}_{cur}$; $k \leftarrow |\mathcal{T}_{cur}|$
15 $\quad\quad r_j \leftarrow$ RAND$(0 \ldots k-1)$
16 $\quad\quad$ **for** j *in* $0 \ldots k$ **do**
17 $\quad\quad\quad next' \leftarrow (r_j + j)$ mod k
18 $\quad\quad\quad (\mathcal{T}_{cur}, \overrightarrow{tv}_{cur}) \leftarrow$ COMPACTION$(\mathcal{T}_{cur}, next')$
19 $\quad$ **if** computeOBJ $(\overrightarrow{tv}_{cur}, \overrightarrow{tv}_{best}$, criteria$)$ **then**
20 $\quad\quad$ // $\overrightarrow{tv}_{cur}$ improves over $\overrightarrow{tv}_{best}$ for a given criteria
21 $\quad\quad \overrightarrow{tv}_{best} \leftarrow \overrightarrow{tv}_{cur}$; $\mathcal{T}_{next} \leftarrow \mathcal{T}_{cur}$;
$\quad\quad packedAPs \leftarrow packedAPs \bigcup \mathsf{tx}(p_{next})$
22 **return** $\mathcal{T}_{next}$;
23 **Procedure** COMPACTION $(\mathcal{T}, i)$**:**
24 $\quad \overrightarrow{tv}_{bestlocal} \leftarrow 0$; $\mathcal{T}' \leftarrow \mathcal{T}$
25 $\quad$ **foreach** $(f, w) \in \mathcal{F}$ **do**
26 $\quad\quad \mathcal{T}[i] \leftarrow (t_i, r_i, f, w)$
27 $\quad\quad \overrightarrow{tv}_{cur} \leftarrow$ estimateTput$(\mathcal{T})$
28 $\quad\quad$ **if** computeOBJ $(\overrightarrow{tv}_{cur}, \overrightarrow{tv}_{best}$, criteria$)$ **then**
29 $\quad\quad\quad \overrightarrow{tv}_{bestlocal} \leftarrow \overrightarrow{tv}_{cur}$; $\mathcal{T}' \leftarrow \mathcal{T}$
30 $\quad$ **return** $(\mathcal{T}', \overrightarrow{tv}_{bestlocal})$;

6 Implementation Aspects

Our implementation of FLUIDconsists of: 1) a central controller that generates the conflict graph and uses the RaC-Pack algorithm to schedule packets. We have implemented this on a Linux PC (3.33 GHz dual core Pentium IV, 2 GB DRAM) (about 3,500 lines of C code and a few hundred lines of Perl scripts). 2) Soekris-based wireless APs and clients, modified to implement channel and width switching functionality. The scheduler is a kernel module that utilizes *high-resolution* timers. In order to reduce communication path latencies, we have implemented a

direct path between the Ethernet and WiFi drivers for the APs. This allows packets received on the wired interface to be immediately forwarded to the wireless interface, bypassing the kernel network queue. We also made driver modifications to ensure that transmit buffers are not flushed, and that clients do not disassociate with the AP when switching frequencies or widths. We now highlight some of the other implementation aspects and system design issues that arise when deploying FLUID.

Handling uplink transmissions. To account for uplink (client-to-AP) transmissions, we use a two-phase TDMA approach [16], [18]: the first phase uses flexible channelization for downlink traffic, and the second phase is for uplink traffic using DCF . The controller *adapts* the time for each phase according to the downlink/uplink traffic ratio (based on queue lengths). By default, since most traffic in enterprise WLANs is downlink [26], we use a 4:1 ratio between the downlink and uplink phases. Carrier sensing and ACKs are disabled in the downlink phase, since they add overheads in a TDMA MAC [16]. Instead, we use block ACKs that are transmitted in the uplink phase. FLUID controller uses this feedback to schedule retransmissions and to refine the modeled conflict graph. To assign channel widths and frequencies in the uplink phase, we use a simple approach: each AP groups its clients into one of four channel widths, based on the widest channel width each client can successfully communicate on. During the uplink phase, FLUID APs switch to their respective center frequencies, and operate on one of the channel widths; over time, the APs cycle through all channel widths with average dwell times at each width being proportional to aggregate uplink traffic from each group. We realize that an optimal assignment for the uplink phase is a challenging problem, and are actively investigating solutions to this problem.

Association. APs are modified to beacon at the lowest channel width of 5 MHz, which has the most range. The center frequencies for beacon transmissions are decided using RaC [5], a conflict-aware fixed-width channel assignment mechanism. Client drivers are modified to perform passive scans using a width of 5 MHz. In our current implementation, we do not support active scanning.

Coordinated switching. To inform the clients about their future schedules, APs use the 802.11 Beacon Information Element (BIE). BIE consists of a list of [*epoff, phase, chan, clist*] where *epoff* is the epoch offset, *phase* indicates uplink or downlink, *chan* is the frequency and width, and *clist* is the list of clients for which traffic has been scheduled in the epoch. To account for beacon losses, the APs also insert a layer 2.5 header in the data packets with information about future schedules. We use built-in Atheros clock synchronization to synchronize the epoch boundaries at APs and the clients.

Implementation overheads. We instrumented the drivers to calculate the delays in controller-AP-client communication path and channel/width switching. We observed that the overheads are dominated by the channel and width switching component; the mean/std. deviation for which was 4.11/0.244 ms. To amortize these overheads, 1) we set the epoch duration to 6 ms, and 2) we use two interfaces at the APs. While one interface is active during an epoch (i.e., it is involved in communication), the other interface prepares for the next epoch. These switching overheads could reduce in future; emerging wireless cards have switching latencies of less than $100\ \mu s$ [1], while prior work in solid state electronics has shown that this delay can be reduced to as low as $40\ \mu s$ [9]. Finally, in order to maintain an accurate conflict graph that can take into account the dynamics of the environment, it is important that the signal strengths are frequently updated. Since there is little external interference in our experimental testbed, which is also likely in other enterprise networks, we chose a measurement periodicity of 10 seconds. However, this is a tunable parameter, and in a more noisy environment one could reduce the measurement periodicity. Similar to previous systems like DIRC [16] and CENTAUR [26], each measurement instance in FLUID lasts for 4 ms. We note that the results in Section 7 include these measurement overheads.

7 Evaluation

Our testbed evaluation aims at characterizing the throughput improvements with FLUID and demonstrate its feasibility on commodity 802.11 hardware. We first evaluate FLUID over a large number of canonical topologies to systematically characterize the performance gains that stem from different components. We show the results for both max-throughput (FLUID-thr) and best min-max throughput (FLUID-fair). Next, we evaluate FLUID over a 23 node representative topology and quantify the performance gains. We perform the experiments at different fixed PHY rates and with dynamic rate adaptation. When using rate adaptation, we run DCF and CENTAUR using SampleRate, and for FLUID, we use the SINR based rate adaptation mechanism (Section 5). We assume that a total of 40 MHz spectrum is available. We quantify the gains of FLUID over DCF with fixed channel width configurations, i.e., 1) DCF using a single 20 or 40 MHz channel (DCF-20 or DCF-40) and 2) DCF using two 20 MHz channels and RaC-based channel assignment [5], denoted by DCF-2x20. To understand the gains attributable to flexible channelization (i.e., variable channel widths and packing) alone, we also compare with DCF employing flexible channelization (DCF-flex) and CENTAUR, a fixed channel width centralized scheduling (TDMA) approach which can exploit exposed terminals [26]. In our experiments, we operate CENTAUR at 40 MHz. The traffic on all the links is backlogged. We report the aggregate throughput in each case, and use Jain's Fairness Index [13] to report overall fairness. Table 4 summarizes the results presented in the paper.

7.1 Gains from Using Client-Centric Widths

FLUID improves the throughput by using client-centric, link quality width aware assignment (e.g., case E1 in Section 2). To evaluate the gains from this aspect, we experiment with 241 single AP-two client topologies with both the clients having SNRs that differ by at least 3 dB. When experimenting with different rates, we only considered cases where the delivery probability of both links was greater than 0.9 at 20 MHz.

- *Different PHY rates.* Table 5 shows that FLUID-thr and FLUID-fair achieve median throughput gains

TABLE 4
Summary of the Results

Section	Evaluation component and set up	Summary of results
Microbenchmarking the model		
§ 4 (Fig. 7, Fig. 6(b))	Accuracy of Delivery Models (231 link pairs, different PHY rates)	RMSE: 14.2%, 8.7%, 8.9%, and 9.6%
§ 4 (Fig. 9)	Packing Accuracy (500 link-intf comb., different PHY rates)	Accuracy: 87.6% ($\mathcal{I}_{t,r}$), 52% (naive)
§ 7.3 (Fig. 11)	Gains w/ packing (331 2-link topologies, Rate: Auto/different PHY rates)	Median gain (Thr./MHz): 1.51× (best DCF)
Gains in specific scenarios		
§ 7.1 (Fig. 10(a), Tab. 5)	Clients with differing SNRs (241 1 AP-2 client topologies, Rates: Auto/different PHY rates)	Up to 1.68×, Median: 1.40× (best DCF), 1.38× (CENTAUR)
§ 7.2 (Fig. 10(b), Tab. 6)	Clients under interference (194 2-link topologies, Rates: Auto/different PHY rates)	Up to 2×, Median: 1.35× (best DCF), 1.32× (CENTAUR)
§ 7.4 (Fig. 12(a))	Hidden links (unscheduled Vs. scheduled) (346 2-link topologies, Rates: Auto/different PHY rates)	Median: 1.74× (best DCF), Avg: 1.88× (DCF-flex), up to 1.47× (CENTAUR)
§ 7.4 (Fig. 12(b))	Exposed links (unscheduled Vs. scheduled) (346 2-link topologies, Rates: Auto/different PHY rates)	Up to 1.97× (best DCF), Median: 1.51× (best DCF), 1.51× (DCF-flex), 1.31×(CENTAUR)
Gains on a representative topology		
§ 7.5 (Fig. 13, Tab. 7)	UDP throughput (23-node topology, Rate:Auto/different PHY rates)	Median: 1.59× (best DCF), 1.34× (CENTAUR)
§ 7.5 (Fig. 13)	TCP throughput (23-node topology, Rate:Auto)	Mean: 1.63×(best DCF), 1.33×(CENTAUR)
§ 7.5	Performance of Rac-Pack (23-node topology, Rate:Auto)	0.95× (brute force), 0.98× (Rac-Pack w/ actual CG)

Gain is reported for throughput unless otherwise noted.

of 44 and 26 percent over the best DCF configuration (DCF-20 or DCF-40), and 41 and 27 percent over CENTAUR across different PHY rates. CENTAUR and DCF-40 do not perform well, as the throughput of the lower SNR client suffers when using a 40 MHz channel. Although DCF-20 improves the SNR by operating the links at 20 MHz, the overall throughput reduces due to spectrum wastage. FLUID operates the higher SNR link at 40 MHz, and the lower SNR link at 20 MHz, with the AP switching between these two widths. FLUID-fair provides lesser gains in order to improve fairness. Fairness indices [13] for FLUID-thr and FLUID-fair were 0.9 and 0.99, while those for DCF-40 and CENTAUR were 0.56 and 0.99.

- *Rate adaptation.* Fig. 10 (left) shows the CDF of throughput gains for CENTAUR and FLUID over the best DCF configuration. We observe that FLUID-thr and FLUID-fair can improve the aggregate throughput up to 68 and 55 percent over DCF, respectively, while improving fairness. The median gains over DCF were around 40 and 25 percent, while those over CENTAUR were 38 and 23 percent. The individual throughput gains for the lower SNR link in these cases were much higher.

TABLE 5
Median Gains (from Using Client-Centric Widths) over Best DCF Configuration and CENTAUR

Scheme	Gains over best DCF config. 12 Mbps	36 Mbps	54 Mbps	Gains over CENTAUR 12 Mbps	36 Mbps	54 Mbps
FLUID-thr	1.43×	1.39×	1.47×	1.41×	1.42×	1.46×
FLUID-fair	1.28×	1.21×	1.26×	1.23×	1.27×	1.25×

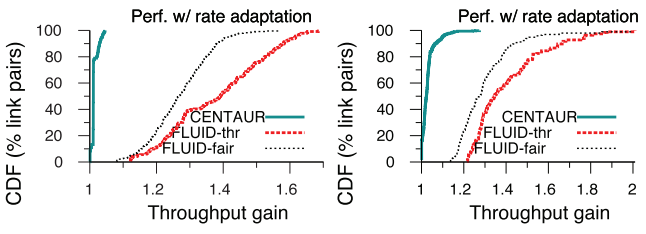

Fig. 10. Throughput gains with rate adaptation for CENTAUR, FLUID-thr and FLUID-fair over DCF with fixed channel widths from (left) link quality aware width assignment (241 single AP—two client topologies). (right) increased transmission concurrency (194 two-link topologies with varying degrees of conflict).

7.2 Gains under Interference

FLUID improves the network throughput by choosing widths which result in increased transmission concurrency under interference (e.g., case E3 in Section 2). To illustrate this, we experiment with 194 one-way hidden interference cases.

- *Different PHY rates.* Here, DCF-40 is unable to resolve the conflict and performs poorly. DCF-2x20 resolves it by assigning the links different channels, whereas CENTAUR does so by serializing the transmissions. However, in many cases, *narrowing the channel width resolves the conflict* due to increase in SINR. FLUID-thr always operates the interfering link at 40 MHz and the other link at 20 MHz, thus allowing *simultaneous* transmissions. To achieve better fairness, FLUID-fair periodically reserves an epoch for the interfered link. Table 6 shows that FLUID-thr and FLUID-fair achieve consistent gains (up to 49 and 29 percent) across different PHY rates due to increased transmission concurrency.
- *Rate adaptation.* Fig. 10 (right) shows that with rate adaptation, FLUID provided up to 2× gains over the best DCF configuration. Median gains for FLUID-fair and FLUID-thr were 28 and 35 percent. Corresponding gains over CENTAUR were 26.2 and 32 percent. In some cases, CENTAUR performs better than DCF-2x20 as operating the links on two

TABLE 6
Median Gains under Interference across Different PHY Rates for FLUID-thr and FLUID-Fair over the Best DCF Configuration and CENTAUR for 194 Topologies

Scheme	Gains over best DCF config. 12 Mbps	36 Mbps	54 Mbps	Gains over CENTAUR 12 Mbps	36 Mbps	54 Mbps
FLUID-thr	1.41×	1.47×	1.49×	1.32×	1.39×	1.46×
FLUID-fair	1.23×	1.29×	1.26×	1.2×	1.26×	1.27×

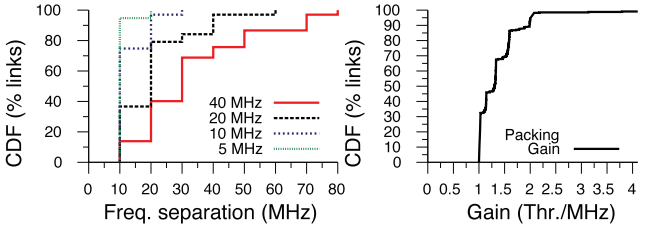

Fig. 11. (left) Minimum center frequency separation required for conflict resolution at different widths. (right) CDF of gain from intelligent packing across 331 link pairs.

adjacent 20 MHz channels was not enough to reduce the conflict.

7.3 Gains from Conflict-Aware Packing

FLUID's gains also stem from 1) efficient transmission packing using partial spectral overlaps and 2) avoiding harmful packing by separating the center frequencies by at least f_{min} (Section 4). We experimented with 331 two-link topologies, where both the links were using the same channel width w. Fig. 11 (right) shows that f_{min} for these links varies—some links benefit from efficient packing ($f_{min} < w$) and others need greater frequency separation ($f_{min} > w$). Fig. 11 (right) shows the CDF of packing gain in terms of throughput per unit MHz. For links with $f_{min} > w$, a maximum gain of $4\times$ (with median 51 percent) was observed as DCF suffered from losses due to interference. For links with $f_{min} < w$, efficient packing resulted in gains up to 70 percent.

7.4 Unscheduled and Scheduled Approaches

To compare scheduled and unscheduled approaches employing flexible channelization, we experiment on 346 two-link topologies that fall into one of two categories (when using 40 MHz): 1) conflicting links or hidden terminals, 2) nonconflicting links and exposed terminals. We compare FLUID with three schemes: 1) DCF-fixed: this is the best among all DCF configurations where both the links use the same channel width 2) DCF-flex: this is the best among all DCF configurations where links can use any permissible combination of channel widths and frequencies, and 3) CENTAUR operating on single channel of 40 MHz.

- *Conflicting/hidden links.* Fig. 12 (left) shows the results for cases where DCF-fixed is unable to resolve the hidden interference. DCF-flex provides significant throughput gains over DCF-fixed (e.g., 63 percent at 12 Mbps and 56 percent at 54 Mbps). However, DCF-flex alone is unable to resolve the conflicts for many other links (e.g., 58 percent of the links at 54 Mbps). CENTAUR is able to resolve all conflicts by virtue of scheduling. Interestingly, DCF-flex performs better than CENTAUR for a certain fraction of links (e.g., 21 percent of the links at 54 Mbps with median gain of 33 percent)—variable channel widths help resolve the conflict, allowing the links to transmit simultaneously. FLUID performs the best (median gain of 74 percent over DCF-fixed) by using scheduling at 40 MHz when it is not possible to resolve conflicts, and using variable channel widths otherwise, to achieve the maximum throughput.
- *Nonconflicting and exposed links.* Here, CENTAUR and FLUID, both exploit the exposed terminals available at 40 MHz in our topologies, e.g., at 12 and 54 Mbps, throughput for 44 and 21 percent of the links is improved by up to $2\times$ (Fig. 12 (right)). The median gain for these exposed terminals was 47 percent. FLUID performs better than CENTAUR, as it exploits the *additional* exposed terminals that arise when using a combination of different channel widths. At 12 and 54 Mbps, the median gain over CENTAUR for these links was 34 and 42 percent. Fig. 12 (right) also shows that FLUID is particularly useful at higher rates, as the number of exposed links available when using only 40 MHz are reduced.
- *Rate adaptation.* With rate adaptation, the median throughput gain of DCF-flex over DCF-fixed was 34 percent across different conflicting link scenarios. FLUID was able to resolve all conflicts, and for exposed links, we observed gains of up to $1.97\times$, with median gains of 51 percent over DCF-fixed and 31 percent over CENTAUR. We note that the gains were much higher in the presence of hidden links when using SampleRate because the links fall back to a lower rate in case of DCF-fixed, while FLUID can continue to operate at a higher rate.

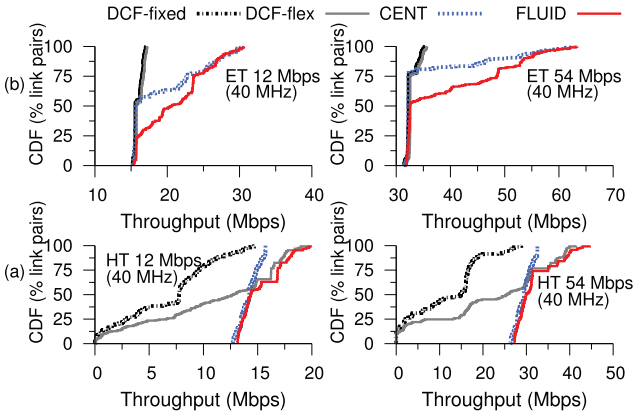

Fig. 12. The plot shows the CDF of throughputs at 12 and 54 Mbps (40 MHz bandwidth) for two categories: (left) conflicting links or hidden terminals. (right) nonconflicting links and exposed terminals. We experimented with 346 two-link topologies for different configurations: 1) DCF-fixed, 2) DCF-flex, 3) CENTAUR, and 4) FLUID.

7.5 Performance on a Representative Topology

We evaluate FLUID on a representative topology by emulating the structure of in-building WLANs. We place our testbed APs near the production APs and clients are randomly distributed into offices without any bias. Our topology consists of 8 APs and 15 clients. For FLUID, we use the modeled conflict graph (Section 4). We also compute the actual conflict graph using bandwidth tests [12] at all possible frequencies and widths. We assume that a total of 40 MHz is available and compare FLUID with DCF-2x20, DCF-40, and CENTAUR. The uplink traffic load is 20 percent of the downlink. Throughput numbers are averaged over 15 runs. Unless otherwise stated, experiments are run using rate adaptation.

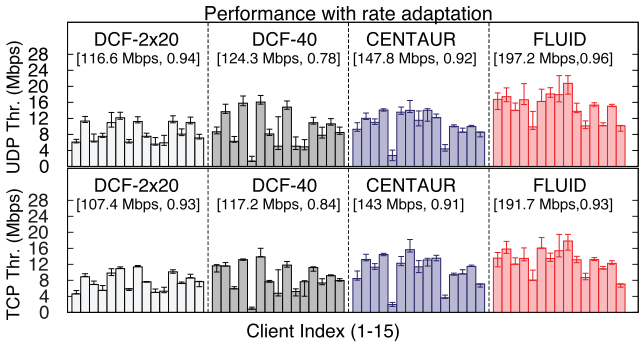

Fig. 13. Throughput achieved with rate adaptation for a 23 node (8 AP, 15 Client) topology. Plot shows the UDP throughput (top) and the TCP throughput (bottom). Tenth and 90th percentile values shown by error bars. Sum of values, Jain's Fairness are shown in parenthesis.

- *UDP throughput.* Fig. 13 (top) shows the UDP throughput for different schemes with rate adaptation. The overall gain was 59 percent over the best DCF configuration (DCF-40) and 34 percent over CENTAUR. FLUID significantly improves throughputs of clients which have a lower SNR at 40 MHz (clients 5 and 11), avoiding harmful packing (clients 8 and 9), and increasing transmission concurrency by exploiting partial overlaps and using variable widths whenever possible (e.g., clients 2 and 6). The 10th, 50th, and 90th percentile gains over DCF in this case were $2.66\times$, $1.54\times$, and $1.21\times$.
- *TCP throughput.* In this experiment, we run bi-directional TCP traffic with 80:20 downlink/uplink split. Fig. 13 (bottom) shows the TCP throughputs for each scheme with rate adaptation. We observe that the average gains over DCF-40 and CENTAUR were 63.5 and 34 percent.
- *Performance of RaC-Pack algorithm.* We evaluated the performance of RaC-Pack algorithm used in FLUID for this topology by comparing it with the brute-force approach which picks the "best" set of schedules, after evaluating all possible schedules offline. We also evaluate the performance of RaC-Pack with the actual conflict graph as input. The aggregate UDP throughputs for these approaches were 206.2 and 201.4 Mbps, respectively, confirming the accuracy of conflict graph and efficiency of the packing approach.
- *Different PHY rates.* Table 7 shows throughput gains of FLUID over the best DCF configuration and CENTAUR for different PHY rates. We observe consistent gains across PHY rates (57-71 percent over best DCF, and 31-58 percent over CENTAUR) on this topology as FLUID was able to successfully resolve the conflicts, and exploit the exposed links available with variable channel widths and partial spectral overlaps.

8 Related Work

Flexible channelization. Recently researchers have explored mechanisms that assign fine grained spectrum blocks on the basis of traffic demands. However, most of these mechanisms, such as DSAP [7], Jello [15], SWIFT [24], WhiteFi [6], ODS [20] and DIMSUMnet [17] are designed for non-802.11 systems and focus on deriving elegant algorithms for spectrum allocation that are primarily evaluated using network simulations or small-scale prototype implementation on software defined radios. FLUID on the other hand is stylized to 802.11 standard in which the rules of carrier sensing define interference in a certain way leading to hidden and exposed terminals. Moreover, FLUID is evaluated using large scale experiments on a 50 node in-building wireless testbed equipped with off the shelf 802.11 hardware.

TABLE 7
Normalized Throughput Gains of FLUID over the Best DCF and CENTAUR across Different PHY Rates

PHY Rate	Gains over best DCF config.			Gains over CENTAUR		
	10th pc	50th pc	90th pc	10th pc	50th pc	90th pc
Fixed 6 Mbps	2.41×	1.62×	1.21×	2.17×	1.31×	1.09×
Fixed 12 Mbps	2.23×	1.63×	1.28×	2.04×	1.34×	1.05×
Fixed 36 Mbps	2.37×	1.57×	1.11×	2.73×	1.46×	1.12×
Fixed 54 Mbps	2.94×	1.71×	1.12×	2.87×	1.58×	1.07×

In the context of 802.11 systems, recent work [8] has shown how adapting channel widths can be beneficial when considering a single, isolated link. VWID [21] analytically shows how variable channel widths can improve transmission concurrency, and demonstrates capacity improvements using small scale experiments. Using analysis and simulations, authors in [22] show how channel widths can be used for load balancing. FLUID builds upon a number of these ideas and extends them significantly to build a practical system, capable of leveraging variable width gains under large scale realistic wireless settings. Another approach to changing channel widths is by adding and removing OFDM subcarriers, as in S-OFDMA. We note that our techniques are useful in such networks as well. If the wireless cards emit the same amount of energy irrespective of the width, then our models hold as is. If the energy varies with the number of subcarriers, then the signal interpolation model in Section 4 can be easily modified to scale the transmitted signal with the channel width. Further, FLUID is complementary to recently proposed fine grained frequency division mechanisms like OFDMA [3] and FARA [23], and can be combined with such mechanisms to provide better gains.

Scheduling in enterprise WLANs. CENTAUR [26] proposes using epoch-based scheduling mechanism for enterprise WLANs. FLUID also uses similar epoch-based scheduling mechanisms to allocate flexible channels on a per-epoch basis. More recently, centralized scheduling has also been used in the context of directional antennas [16] and MIM-aware transmission reordering [18]. Authors in [16], [18] also propose using a two-phase TDMA approach similar to FLUID to accommodate uplink traffic.

Conflict graphs. Recent research has used SINR-based models to efficiently generate conflict graphs for fixed-width systems [4], [14], [16]. However, existing methodologies to generate conflict graphs are not suited for use with flexible channelization as they incur significant overhead. We build upon their work, and use SINR-based delivery models to predict interference between links using variable channel widths while allowing arbitrary spectral overlaps.

9 Conclusion

In this paper, we explored the opportunities and challenges in designing 802.11-based wireless LANs employing flexible channelization. We demonstrated that while flexible channelization can improve system throughput, careful construction of flexible channels requires taking into account the interference parameters of the network that depend on the combination of frequencies and channel widths, topology, and traffic demand. To this end, we designed and implemented FLUID, a system which improves the throughput of enterprise WLANs by employing joint flexible channelization and data scheduling. Testbed results demonstrate the feasibility of our approach and throughput improvements show that flexible channelization can be a useful parameter in WLAN design.

Acknowledgments

The authors would like to thank the anonymous reviewers whose comments helped bring the paper into its final form. S. Rayanchu, V. Shrivastava, and S. Banerjee were supported in part by the US National Science Foundation through awards CNS-1059306, CNS-0855201, CNS-0747177, CNS-0916955, CNS-1040648, and CNS-1064944.

References

[1] "Intel Pro/Wireless Network Connection for Mobile," http://www.intel.com/network/connectivity/products, 2012.
[2] "MadWifi Wireless Driver," http://madwifi-project.org, 2012.
[3] "WiMAX Forum Whitepapers," http://www.wimaxforum.org, 2012.
[4] C. Reis et al, "Measurement-Based Models of Delivery and Interference in Static Wireless Networks," *Proc. ACM Special Interest Group on Data Comm. (SIGCOMM '06)*, 2012.
[5] A. Mishra et al., "A Client-Driven Approach for Channel Management in Wireless LANs," *Proc. IEEE INFOCOM*, 2006.
[6] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White Space Networking with Wi-Fi Like Connectivity," *Proc. ACM Special Interest Group on Data Comm. (SIGCOMM '09)*, 2009.
[7] V. Brik, E. Rozner, and S. Banerjee, "DSAP: A Protocol for Coordinated Spectrum Access," *Proc. IEEE First Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, 2005.
[8] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "Case for Adapting Channel width in Wireless Networks," *Proc. ACM Special Interest Group on Data Comm. (SIGCOMM '08)*, 2008.
[9] F. Herzel, G. Fischer, and H. Gustat, "An Integrated CMOS RF Synthesizer for 802.11a Wireless LAN," *IEEE J. Solid-State Circuits*, vol. 38, no. 10, pp. 1767-1770, Oct. 2003.
[10] G. Stuber, *Principles of Mobile Communication.* Springer, 2000.
[11] *IEEE Wireless LAN MAC and PHY Specification*, IEEE, 1999.
[12] J. Padhye et al, "Estimation of Link Interference in Static Multi-Hop Wireless Networks," *Proc. Fifth ACM SIGCOMM Conf. Internet Measurement (IMC '05)*, 2005.
[13] R. Jain et al., "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," Technical Report TR-301, Digital Equipment Corp. 1984.
[14] A. Kashyap, S. Ganguly, and S.R. Das, "Measurement-Based Approach to Modeling Link Capacity in 802.11 Wireless Networks," *Proc. ACM MobiCom*, 2007.
[15] L. Yang, W. Hou, B.Y. Zhao, and H. Zheng, "Supporting Demanding Wireless Applications with Frequency-Agile Radios," *Proc. Seventh USENIX Symp. Networked Systems Design and Implementation (NSDI '10)*, 2007.
[16] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "DIRC: Increasing Indoor Wireless Capacity Using Directional Antennas," *Proc. ACM Special Interest Group on Data Comm. (SIGCOMM '09)*, 2009.
[17] M. Buddhikot et al., "DIMSUMnet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," *Proc. IEEE Sixth Int'l Symp. World of Wireless Mobile and Multimedia Networks (WoWMoM '05)*, 2005.
[18] J. Manweiler, N. Santhapuri, S. Sen, R.R. Choudhury, S. Nelakuditi, and K. Munagala, "Order Matters: Transmission Reordering in Wireless Networks," *Proc. ACM MobiCom*, 2008.
[19] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh, "Partially Overlapped Channels Not Considered Harmful," *Proc. Joint Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS '06)*, 2006.
[20] R. Gummadi and H. Balakrishnan, "Wireless Networks Should Spread Spectrum on Demand," *Proc. Seventh ACM Workshop Hot Topics (HotNets '08)*, 2008.
[21] R. Gummadi, R. Patra, H. Balakrishnan, E. Brewer, "Interference Avoidance and Control," *Proc. Seventh ACM Workshop Hot Topics (HotNets '08)*, 2008.
[22] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan, "Load-Aware Spectrum Distribution in Wireless LANs," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '08)*, 2008.
[23] H. Rahul, F. Edalat, D. Katabi, and C.G. Sodini, "Frequency-Aware Rate Adaptation and MAC Protocols," *Proc. ACM MobiCom*, 2009.
[24] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to Share: Narrowband-Friendly Wideband Networks," *Proc. ACM Special Interest Group on Data Comm. (SIGCOMM '08)*, 2008.
[25] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-Grained Channel Access in Wireless LAN," *Proc. ACM Special Interest Group on Data Comm. (SIGCOMM '10)*, 2010.
[26] V. Shrivastava et al., "CENTAUR: Realizing the Full Potential of Centralized WLANs through a Hybrid Data Path," *Proc. ACM MobiCom*, 2009.
[27] L. Yang, L. Cao, H. Zheng, and E. Belding, "Traffic-Aware Dynamic Spectrum Access," *Proc. Fourth Ann. Int'l Conf. Wireless Internet (WICON '08)*, 2008.
[28] Y. Yuan et al., "Allocating Dynamic Time-Spectrum Blocks in Cognitive Radio Networks," *Proc. ACM MobiHoc*, 2007.

Shravan Rayanchu received the master's degree in computer science from the University of Wisconsin-Madison (UW Wisconsin), and the BTech degree in computer science from the Indian Institute of Technology (IIT), Guwahati. He is currently working toward the PhD degree in the Computer Science Department at UW Wisconsin. His research focuses on diagnosing performance issues in wireless networks and improving their performance by designing smart control and data plane mechanisms. He was a recipient of the Microsoft Research PhD fellowship. He won the best paper award for work on hybrid scheduling at MobiCom 2009 and his work on flexible channelization was nominated for the best paper at MobiCom 2011.

Vivek Shrivastava received the PhD degree from the University of Wisconsin-Madison (UW Wisconsin) in 2010 under the guidance of Professor Suman Banerjee. He is a part of Mobile Computer Systems team at the Nokia Research Center, Palo Alto, where he works on advanced smartphone design and prototyping. At UW Wisconsin, he worked on improving wireless connectivity and performance, especially in enterprise settings. His work demonstrated that the inherent centralization in enterprise WLANs can be leveraged to enable high performance, predictable, and manageable wireless networks. He also received the IBM PhD fellowship for 2009-2010 and a best paper award at Mobicom 2009 for his work in hybrid scheduling in enterprise networks.

Suman Banerjee is an associate professor in computer science and electrical and computer engineering at the University of Wisconsin-Madison where he heads the Wisconsin Wireless and NetworkinG Systems (WiNGS) laboratory. He has worked extensively in the domain of mobile and wireless networking systems for the last 10 years. He served as a technical program chair of ACM MobiCom 2010. He has published more than 100 technical papers in his field, and some of his recent successes include a best paper award at ACM MobiCom 2009 and a best paper nomination at ACM MobiCom 2011.

Ranveer Chandra received the undergraduate degree from IIT Kharagpur, India, and the PhD degree from Cornell University in 2005. He is a senior researcher at Microsoft Research. As part of his doctoral dissertation, he developed VirtualWiFi—a virtualization architecture for wireless network cards. This software has been downloaded more than 150,000 times, and is among the top 5 downloaded software released by Microsoft Research. Windows 7 supports VirtualWiFi APIs as well. His research focuses on system challenges in designing computer networks. He is currently working on four different projects: white space networking, energy saving PC architecture, mobile systems, and network management. He was invited to the FCC to present his research on white spaces. Spectrum regulators from India (including TRAI Chairman J.S. Sarma), China (SARFT), Brazil (ANATEL), Singapore (IDA), and US (including FCC Chairman Genachowski) have visited the Microsoft Campus in Redmond, Washington to see a live demonstration of his research. He has published more than 35 research papers and filed more than 40 patents, 10 of which have been granted. He has won several awards, including best paper awards at ACM CoNext 2008 and ACM SIGCOMM 2009, the Microsoft Graduate Research Fellowship, and the MIT Technology Review TR35 (2010).