Rate Adaptation for 802.11 Multiuser MIMO Networks

Wei-Liang Shen†‡, Yu-Chih Tung†, Kuang-Che Lee†, Kate Ching-Ju Lin†,
Shyamnath Gollakota◦, Dina Katabi♯ and Ming-Syan Chen†‡
†Research Center for IT Innovation, Academia Sinica
‡Department of Electrical Engineering, National Taiwan University
◦CSE, University of Washington
♯CSAIL, Massachusetts Institute of Technology

ABSTRACT

In multiuser MIMO (MU-MIMO) networks, the optimal bit rate of a user is highly dynamic and changes from one packet to the next. This breaks traditional bit rate adaptation algorithms, which rely on recent history to predict the best bit rate for the next packet. To address this problem, we introduce TurboRate, a rate adaptation scheme for MU-MIMO LANs. TurboRate shows that clients in a MU-MIMO LAN can adapt their bit rate on a per-packet basis if each client learns two variables: its SNR when it transmits alone to the access point, and the direction along which its signal is received at the AP. TurboRate also shows that each client can compute these two variables passively without exchanging control frames with the access point. A TurboRate client then annotates its packets with these variables to enable other clients to pick the optimal bit rate and transmit concurrently to the AP. A prototype implementation in USRP-N200 shows that traditional rate adaptation does not deliver the gains of MU-MIMO WLANs, and can interact negatively with MU-MIMO, leading to low throughput. In contrast, enabling MU-MIMO with TurboRate provides a mean throughput gain of 1.7x and 2.3x, for 2-antenna and 3-antenna APs respectively.

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

General Terms Algorithms, Design, Performance, Theory

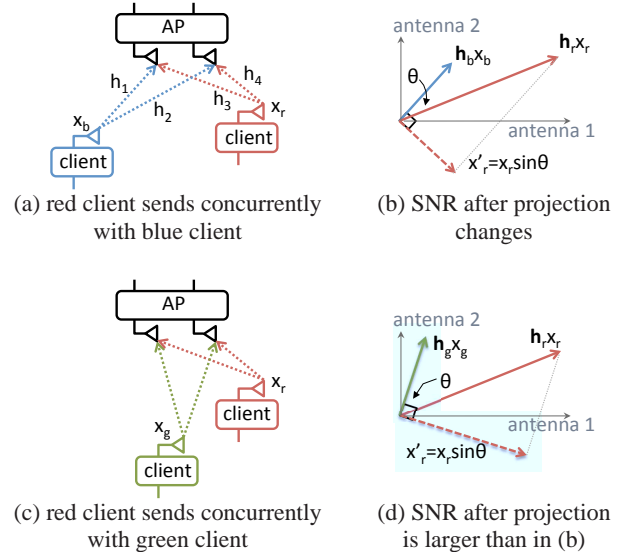**Keywords** Multiuser MIMO networks, Rate adaptation

1. INTRODUCTION

Wireless LANs are facing two trends: First, the number of antennas on an access point is increasing steadily, with typical APs today having two or three antennas [1]. Second, there is a proliferation of small WiFi devices, e.g., sensors, smart phones, and game consoles, which have a small form factor and strict power limitations, and hence typically use a single antenna. These trends cause a multi-antenna access point to spend a significant fraction of its time communicating with a single antenna client. As a result, wireless LANs will not deliver the maximum number of concurrent transmissions enabled by their infrastructure. To address this problem, researchers have advocated the use of multiuser MIMO

(a) red client sends concurrently with blue client

(b) SNR after projection changes

(c) red client sends concurrently with green client

(d) SNR after projection is larger than in (b)

Figure 1—Optimal bitrate changes after projection. The optimal bitrate depends on the client that is transmitting concurrently. The smaller angle between the concurrent clients leads to a larger amount of SNR reduction after projection.

(MU-MIMO) LANs, where multiple single-antenna clients communicate concurrently with a multi-antenna AP. They demonstrated that decoding such concurrent transmissions is feasible both on the uplink and downlink [31, 7, 37]. They also developed a MAC protocol that allows clients to contend for concurrent transmissions to a multi-antenna AP [31]. So far, however, research on MU-MIMO WLANs has not addressed the bit rate selection problem, and simply assumed that the transmitters know the best bit rate [31, 7, 37]. This assumption is valid on the downlink where there is only one transmitter, the AP, and hence the problem can be reduced to standard 802.11n rate adaptation. The scenario on the uplink, however, is quite different: it has multiple concurrently transmitting clients that collectively have to pick the best bit rates to their AP. The decisions made by these clients are not independent; they interact in a complex manner that intrinsically differs from existing 802.11 networks.

To see the problem, consider the scenario in Fig. 1(a) where two single-antenna clients transmit concurrently to a 2-antenna access point. Recall that a 2-antenna AP receives signals in a 2-dimensional space defined by its two antennas, as shown in Fig. 1(b). The basic approach for decoding the concurrent packets is as follows [33]: The AP first projects the incoming signal

on a direction orthogonal to one of the clients, say the blue client. This eliminates the signal of the blue client and allows the AP to decode the red client. The AP then uses interference cancellation to subtract the red client's signal and decode the blue client. Note that the success of this decoding process depends on the AP being able to decode the red client after projecting its signal on a direction orthogonal to the blue client. This projection however reduces the SNR of the red client, as evident from the reduction in the length of the projected red vector in Fig. 1(b). This means that the red client should transmit at a bit rate supported by its SNR after projection; otherwise the AP becomes unable to decode its signal. Note also that the SNR after projection and hence the optimal bit rate depends on the angle between the signals of the two clients, i.e., θ. For example, if the red client transmits its next packet with the green client, as in Fig. 1(c), then its SNR after projecting on a direction orthogonal to the green client will be different, as in Fig. 1(d), and hence the red client's optimal bit rate for the next packet will change.

Thus, in a MU-MIMO LAN, the optimal bit rate of a client changes depending on the set of clients that transmit with it. Since this set may vary from one packet to the next, the optimal bit rate changes on a per packet basis. This breaks the basic assumption underlying existing 802.11 bit rate adaptation algorithms, which use the bit rate that fits recent packets as a predictor for the best bit rate for the next packet [17, 27, 9, 24, 8, 35].

This paper presents TurboRate, a bit rate adaptation protocol suitable for concurrent MU-MIMO 802.11 clients. TurboRate enables a MU-MIMO client to pick the optimal bit rate for each packet it transmits, even when the bit rate changes from one packet to the next.

At a high level, TurboRate works as follows. Each client listens to the AP's transmissions (including its beacons) to learn the channel coefficients from the AP to itself. The client uses this information to *passively* compute two variables: 1) the direction along which the AP receives its signal, and 2) its SNR if it were to transmit to the AP alone (i.e., its SNR without projection). For example, in the case of a 2-antenna AP, the direction along which the AP receives a client's signal can be identified by the direction of its channel vector, e.g., $\mathbf{h}_b = (h_1, h_2)$ for the blue client as shown in Fig. 1(b), and the client's SNR can be computed as $\|\mathbf{h}\|^2 P/N_0$, where $\mathbf{h}$ is the vector of channels that the client passively measures from the AP's transmissions, P is the client's transmission power, and N_0 is the noise level at the AP, which we include in the beacons.[1] When clients contend for concurrent transmissions, the client that wins the contention first starts its packet with a special header that includes the direction along which the AP receives its signal. A client that wants to transmit concurrently with the first client uses this information to project its signal orthogonal to the first client and compute the reduction in its SNR. It then maps its SNR after projection to the optimal bit rates using standard SNR-bitrate tables [17, 27]. Additional concurrent clients can join the transmission and compute their optimal bit rate using the same process.

A notable feature of TurboRate is that it works in a distributed random access manner. Specifically, a client, e.g., the blue client in Fig. 1(a), can win the contention and transmit, picking its bit rate as usual without knowing whether other clients have packets and may transmit concurrently. A client, like the red client, that decides to transmit concurrently with the first client does not have to confer

---

with it; it simply picks a bit rate that does not interfere with the first client's reception.

We built a prototype of TurboRate using the USRP-N200 radio platform and evaluated it over a 10 MHz channel. Our implementation uses an OFDM PHY-layer and supports the various modulations (BPSK, 4-64 QAM) and coding options used in 802.11. Our results are as follows:

- Activating MU-MIMO with existing bit rate selection fails to deliver its gains and can lead to a significant throughput reduction. In particular, we experimented with different client positions that span the range of inter-client reception angle, i.e., $\theta \in [0, \pi/2]$. The results show that, in 90% of the studied cases, enabling MU-MIMO without addressing its special needs for per-packet bit rate adaptation reduces the throughput below that achieved with a single client. Further, in about 50% of the cases, the network throughput reduces to zero because the clients' rates overshoot the capacity of the network.

- TurboRate's bit rate selection enables MU-MIMO to deliver its gains. With TurboRate, MU-MIMO produces an average throughput gain of 1.7x in the case of 2-antenna AP and 2.3x in the case of 3-antenna AP.

TurboRate enables distributed bitrate adaptation for MU-MIMO LANs. The closest to our work is 802.11n$^+$ [23], which supports per packet bit rates, but addresses a different problem in which concurrent clients communicate with different APs. It also assumes that concurrent clients have a different and increasing number of antennas (i.e., one client has a single antenna, the second has two antennas, and the third has three antennas). In contrast, TurboRate can support clients, with the same or different numbers of antennas, transmitting to an AP in a MU-MIMO LAN.

2. UNDERSTANDING RATE SELECTION IN MU-MIMO

Before describing our proposed rate adaptation protocol, we conduct theoretical analysis and testbed measurements to understand how MU-MIMO concurrent transmission changes a client's optimal bit rate and the implications of picking the wrong bit rate. We focus on the scenario in Fig. 1(a), where two single-antenna clients communicate with a 2-antenna AP. The maximum bit rate of both clients is limited by the need to ensure that the access point can still decode the signals. Let x_b be the symbol transmitted by the blue client and x_r be the symbol transmitted by the red client, concurrently. The 2-antenna AP receives the combined signals in a 2-dimensional antenna space, as shown in Fig. 1(b),

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} x_b + \begin{pmatrix} h_3 \\ h_4 \end{pmatrix} x_r + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix},$$

where the vector $\mathbf{h}_b = (h_1, h_2)$ is the channels of the blue client and the vector $\mathbf{h}_r = (h_3, h_4)$ is the channels of the red client in the antenna space, as shown in Fig. 1(b), and n_1 and n_2 are the noise observed at the AP's two antennas. For simplicity, we assume that n_1 and n_2 are independent and follow the same Gaussian distribution $n_1, n_2 \sim \mathcal{CN}(0, N_0)$, where N_0 is the average noise power at the AP.

Say the AP is interested in decoding the red client, x_r. To null out the interfering signal, x_b, the AP uses a technique called *zero-forcing (ZF)* [33] to project the received signal on a direction orthogonal to x_b, i.e., $(h_2, -h_1)$, which can be formalized as follows:

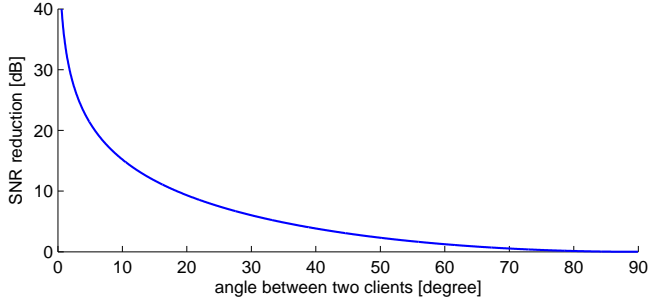$$y_{proj} = h_2 y_1 - h_1 y_2 = (h_2 h_3 - h_1 h_4) x_r + (h_2 n_1 - h_1 n_2).$$

Figure 2—SNR reduction in dB due to projection, as a function of the angle between the two clients at the AP.

Figure 3—Capacity ratio after projection.

It then decodes the projected signal as:

$$x_r' = \frac{y_{proj}}{h_2 h_3 - h_1 h_4} = x_r + n_r = x_r + \frac{h_2 n_1 - h_1 n_2}{h_2 h_3 - h_1 h_4}. \quad (1)$$

We can observe from the above equation that the noise after projection, n_r, is scaled up. The SNR hence decreases after projection, and becomes

$$
\begin{aligned}
SNR_{proj} &= \frac{\mathbb{E}[|x_r|^2]}{\mathbb{E}[|n_r|^2]} \\
&= \frac{|h_2 h_3 - h_1 h_4|^2}{\|(h_2, -h_1)\|^2 N_0} \mathbb{E}[|x_r|^2] \\
&= \frac{|(h_2, -h_1) \cdot (h_3, h_4)|^2}{\|(h_2, -h_1)\|^2 \|(h_3, h_4)\|^2} \frac{\mathbb{E}[\|(h_3, h_4)x_r\|^2]}{N_0} \\
&= \cos^2(\pi/2 - \theta)SNR_{orig} \\
&= \sin^2(\theta)SNR_{orig}, \quad (2)
\end{aligned}
$$

where $(\cdot)$ denotes the inner product, θ is the angle between the channels of two clients, (h_1, h_2) and (h_3, h_4), as in Fig. 1(b), and $SNR_{orig} = \mathbb{E}[\|(h_3, h_4)x_r\|^2]/N_0$ is the SNR of x_r when the red client transmits alone, i.e., without projection. Geometrically, we can see from Fig. 1(b) that the amplitude of x_r after projection is reduced to $sin(\theta)x_r$, matching the above derivation that SNR_{proj} equals $\sin^2(\theta)SNR_{orig}$. The amount of SNR reduction for the red client in dB due to projection orthogonal to the blue client can be expressed as:

$$
\begin{aligned}
\Delta SNR &= 10\log_{10}(SNR_{orig}) - 10\log_{10}(SNR_{proj}) \\
&= -20\log_{10}\sin(\theta). \quad (3)
\end{aligned}
$$

We note two important points:

- First, the direction along which a client is received is defined by its channel vector at the AP. In our example, the blue client is received along the direction (h_1, h_2), and the red client is received along the direction (h_3, h_4). Thus, the angle between two clients, θ, is in the antenna space, not the I-Q plane [15, 13]. Hence, this angle does not change with signal rotation in the complex I-Q plane.
- For general scenarios where a client communicates with an M-antenna AP in the presence of k concurrent transmissions ($k < M$), we can still compute the SNR reduction of this client based on Eq. (3). The only difference is that the AP needs to decode by projecting along the direction orthogonal to all the k concurrent transmissions. In this general case, θ hence becomes the angle between the client and the k-dimensional subspace S spanned by the k concurrent transmissions in the AP's M-dimensional an-
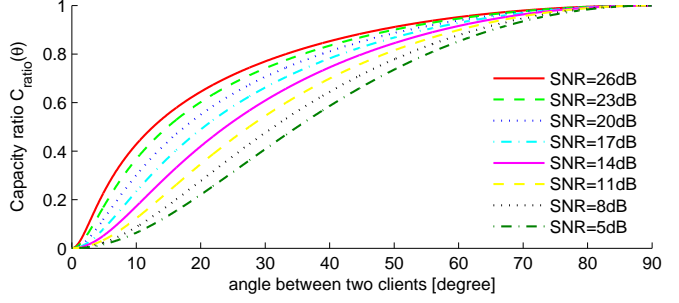
tenna space. The value of $\sin\theta$ can be computed by

$$\sin\theta = \frac{|\mathbf{h}_\perp \cdot \mathbf{h}|}{\|\mathbf{h}_\perp\|\|\mathbf{h}\|}, \quad (4)$$

where $\mathbf{h}$ is the channel vector of the client that we want to decode and $\mathbf{h}_\perp$ is the vector that is orthogonal to the subspace spanned by the k concurrent transmissions[2].

(a) How does zero-forcing affect the SNR of the signal? We can see from Eq. (3) that the reduction in SNR due to zero-forcing is independent of the original SNR of the client, and solely depends on the angle between the clients. Fig. 2 plots the reduction in SNR as a function of the angle between the two clients. It shows that, when the angle is smaller than 45 degree, the SNR reduction exceeds 3 dB. A reduction in SNR larger than 3 dB requires an 802.11 node to reduce the transmission bit rate at least one bit-rate lower [27]. Depending on the actual value of the SNR reduction, it might be insufficient to just go down one bit rate lower. In fact, if the reduction in SNR is such that the SNR after projection is less than 4 dB, a client will be unable to use even the lowest 802.11 bit rate and hence should not transmit concurrently with the ongoing client.

(b) How does the SNR reduction impact the optimal bit rate? Even though the SNR reduction is independent of the original SNR, the change in the optimal bit rate depends on the original SNR. Since the optimal bit rate tends asymptotically to the capacity, we estimate the change in the optimal bit rate as the change in the capacity before and after projection. The ratio of the capacity after projection to the original capacity can be formulated as a function of the angle between the two clients' signals at the AP as follows:

$$C_{ratio}(\theta) = \frac{B\log_2(1+SNR_{proj})}{B\log_2(1+SNR_{orig})} = \frac{log(1+\sin^2(\theta)SNR_{orig})}{log(1+SNR_{orig})},$$

where B is the bandwidth of the channel, and θ is the angle between the two clients at the AP.

Fig. 3 plots the capacity ratio in different SNR_{orig} regimes. The figure shows that, for a particular angle, e.g., $\theta = 30$ degree, a link with a low original SNR experiences a larger capacity drop than that with a high original SNR. It means that the low SNR regime is more sensitive to SNR reduction, and will likely require decreasing the bit rate to support concurrent transmissions. The figure also shows that the median capacity reduction, i.e., the reduction corresponding to an angle of 45 degree, is about 30%. This means that, assuming the distribution of the angle between two clients is uniform over all angles, one would expect the throughput of two concurrent clients in a MU-MIMO to be about 1.7x the throughput

[2]The orthogonal vector $\mathbf{h}_\perp$ can be computed by $\mathbf{h}_\perp = \mathbf{h} - proj_S(\mathbf{h})$, where $proj_S(\mathbf{h})$ is the projection of $\mathbf{h}$ onto the subspace S.
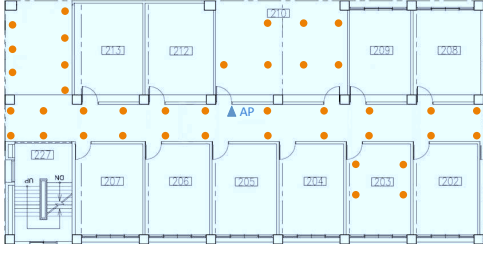
Figure 4—The testbed. Orange dots refer to client locations. Blue triangle refers to the location of the AP.

Figure 5—Throughput gain in TurboRate.

Figure 6—Angles between two clients in real channels.

of a single client transmitting to the same 2-antenna AP. We will see in §6.2 that the median throughput gain in TurboRate is 1.7x for 2-antenna AP scenarios, which shows that TurboRate matches the expected theoretical performance of MU-MIMO.
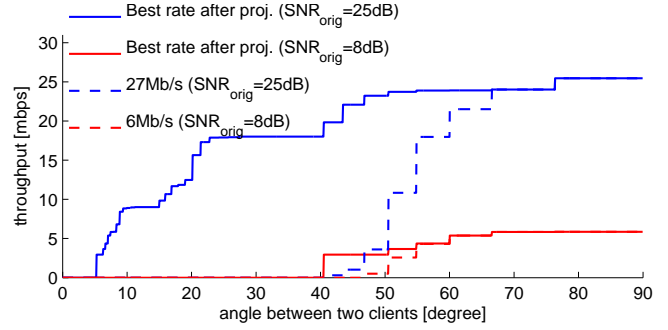
(c) What are the implications of ignoring MU-MIMO in rate adaptation? The above argument shows that the channel capacity of a client changes when it joins a concurrent transmission because the channels of the two clients interact together. The client should react to that change in capacity by adopting a different bit rate than it would adopt if it were transmitting alone. If the client does not react then it might exceed the capacity of its channel leading to its packets becoming undecodable. This also impacts all other clients that are transmitting concurrently, because the aggregate rate of all clients exceeds the combined channel capacity. As we argued earlier, these client channels are not independent from each other; they are related by the angle between the directions along which the AP receives them.

To illustrate this point, we collect empirical measurements using USRP-N200 [2] on a 10 MHz channel. We use the testbed in Fig. 4. We fix the location of the 2-antenna AP, and vary the locations of the two clients. We empirically measure the packet delivery ratio for different bit rates in the entire 802.11 operational range, and compute the throughput by multiplying the rate by the packet delivery ratio corresponding to the SNR after projection.
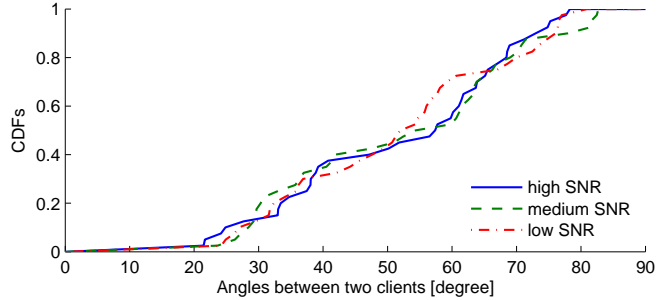
We plot in Fig. 5 the throughput of a client whose original optimal bit rate is 27 Mb/s (i.e., 64-QAM, 3/4 coding rate on a 10 MHz channel) if it were to transmit alone. The dotted blue line shows the throughput of this client if it does not change its rate as a function of the angle between its signal at the AP and that of the concurrent client. The solid blue line is the throughput of the client if it reacts by changing its rate to take into account the angle between its channel and that of the concurrent client, and the resulting SNR reduction. The figure shows that if the client does not change its bit rate, then for any angle smaller than 38 degree, it will get zero throughput. This is because the original bit rate significantly exceeds the capacity of its channel after projection. In contrast, if it does adapt, then it can continue enjoying a significantly higher throughput even for small angles.

For comparison, we also plot in red the same graphs for a low SNR client whose original optimal bit rate if transmitting alone is 6 Mb/s (QPSK, 1/2 coding rate). Note that this client will get a zero throughput for any angle smaller than 40 degree, even if it reduces its bit rate to the lowest rate (i.e., the solid red line) for concurrent transmissions. Thus, a client whose optimal bit rate when transmitting alone is 6 Mb/s should check the angle it has with the other client who has proceeded it to transmit, and if the angle is smaller than 40 degree, it should abstain from contending for the channel.

(d) What are the practical values for the angle between the signals of two clients at a shared AP? The analysis so far assumes

that the angle between the two clients ranges from 0 to 90 degrees. We next use empirical measurements to check the distribution of the angle between the channels of two clients. Again the measurements are conducted using USRP-N200 [2] in the testbed shown in Fig. 4. We fix the location of the 2-antenna AP, and vary the locations of the two clients. We collect measurements for 100 different choices of clients' locations, picked at random from Fig. 4.

Fig. 6 plots the CDFs of the angle between the directions along which the two clients are received. The CDFs are taken over different client locations. The figure shows that the angles are uniformly distributed between 20 and 80 degree in all SNR regimes. Note that an angle of 90 degree shows that the two clients are received orthogonal to each other at the AP and hence their channels do not interact. In contrast, a small angle means that the signals of the two clients interfere significantly and the total capacity is far from the sum of the two capacities. Since the empirical results show that the angle can take a wide range of values, the client has to measure this angle and react appropriately.

3. TURBORATE

TurboRate addresses rate selection on MU-MIMO uplinks. We consider a MU-MIMO MAC protocol similar to SAM [31], where clients contend for concurrent transmissions and join the ongoing transmissions one after another (see [31] for details). In such an MU-MIMO MAC, a client that wins the contention needs to select its best bit rate immediately before data exchange. It however has no idea who and how many other clients will win the contention after it, and transmit concurrently with it. For example, say the AP has three antennas; the first client that wins the contention does not know whether other clients might contend and win the second and third concurrent transmission opportunities. Further, the second client that wins the contention knows only about the first client, but does not know whether there will be a third concurrent client.
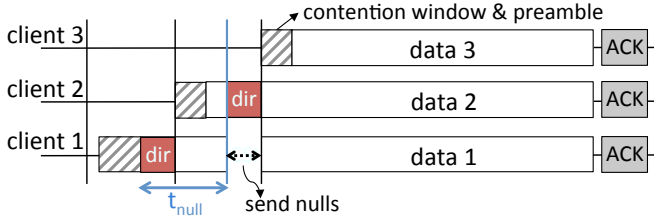
Figure 7—**Rate Adaptation Protocol.** Each client annotates its packets with the direction of its signal at the AP. This information enables potential concurrent clients to select their bit rates. To ensure single-antenna clients can decode these annotations, we force the ongoing transmissions to pause their streams when the contention winner sends the annotated header.

Figure 8—**Amplitude of time-domain channels.** Only first few taps of the 20 MHz 802.11 channel have a noticeable amplitude. To reduce the overhead, a client only announces those taps.

We would like a bit rate adaptation protocol that enables each client to select its bit rate by considering only those clients that won the contention before it, and without worrying about the clients that may win the contention after it.

TurboRate realizes the above goal. At a high level, TurboRate works as follows: Each client passively learns the direction along which it is received at the AP and its SNR if it transmits alone, i.e., SNR_{orig}. During contention, the client learns the direction of other clients that won the contention before it, and uses this information to compute its SNR after projection, SNR_{proj}, and the corresponding optimal bit rate. The AP decodes the concurrent clients using zero-forcing with successive interference cancellation (ZF-SIC) [33].

The next few subsections describe the protocol in detail.

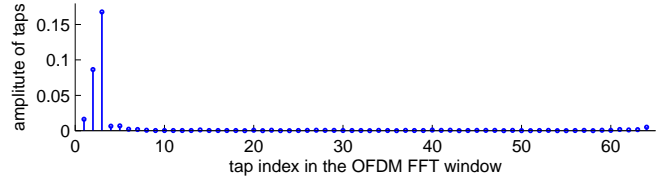### 3.1 Learning a Client's Direction and SNR Passively

TurboRate requires the client to know its own SNR to the AP and the direction along which its signal is received at the AP. Both parameters can be directly derived from the client's channels to the AP. The client's SNR can be easily computed using the preamble [27][29]. As for the direction, a client is received along the direction of its channel vector, i.e., **h**, where the elements of **h** are the channels from the client to the AP's antennas. So, to estimate these variables the client needs to learn its channels to the AP.

One naive mechanism to learn the channels is to have the AP explicitly tell each client its channel values. This solution, however, has a high overhead that increases with the number of clients. In contrast, TurboRate enables the clients to learn their channels to the AP passively by listening to the AP's transmissions including its beacons. Specifically, the clients leverage *channel reciprocity* [16]. Reciprocity refers to the property that the channels in the forward and reverse directions are the transpose of each other because electromagnetic waves travel forward and backward the same way. The feasibility of reciprocity has been verified empirically in [16, 4]. Using reciprocity, every client can exploit the beacons to learn the channels from the AP and estimate the reverse channels. Updating the channels using periodic beacon frames is sufficient because the coherence time of indoor channels is typically between 0.2 second to multiple seconds [33], which is longer than the beacon interval 0.1s. Clients can further refine the estimation opportunistically by overhearing the downlink packets from the AP.

TurboRate also makes the AP measure its noise level and include it in its beacons. Given the channel vector and the AP noise power, each client can estimate its original SNR and the direction along which it is received at the AP.

3.2 Exchanging the Channel Directions

To compute the best rate, a TurboRate client has to consider the SNR reduction after projecting along the direction orthogonal to all the ongoing transmissions. The SNR reduction after projection depends on the angle between its signal and all the ongoing transmissions. To compute this angle, a TurboRate client not only needs to know its own channels, but also the directions of all the ongoing transmissions. A client however can only learn its own channels from the beacons. To enable the client that joins later to know the direction of the ongoing transmissions, each TurboRate client that wins contention announces its channel direction by annotating the PLCP header. Clients that later contend for concurrent transmissions use this information to select their rates.

This simple solution addresses the problem in a 2-antenna AP scenario because all clients can hear the information sent by the first contention winner. This solution, however, does not generalize to more than two antennas. To see this, lets consider a 3-antenna AP that can support up to three concurrent transmissions. While all clients can hear the PLCP header of the first winner, single-antenna clients are unlikely to successfully decode the second client's header in the presence of the first client's ongoing transmission. Thus, to decode this header information, TurboRate forces all the clients to stop transmitting when a client broadcasts its direction. In particular, TurboRate forces clients with ongoing transmissions to pause their streams and send null samples for a period of time that is long enough for the new client to broadcast its direction information.

The issue, however, is that ongoing transmitters do not know when will a new client win contention and broadcast its direction. For example, the client that first wins contention and starts transmitting does not know when the second client wins contention and broadcasts. To avoid this uncertainty, as shown in Fig. 7, in TurboRate the first winner always pauses its transmission at a pre-defined timeslot t_{null} after it wins the contention. In particular, as soon as the second client wins contention, it transmits its preamble[3], stays idle, and then broadcasts its direction in the clear when the first winner keeps silent at t_{null}. More generally, in a network support M concurrent transmissions, TurboRate forces all the ongoing clients to pause their transmissions at times $k * t_{null}$, for all $k = 1, \cdots, M-2$, after the first client wins.

Enabling this protocol however requires the client that wants to join the concurrent transmission to win the contention before t_{null} because the information has to be sent by the winner at t_{null} exactly. To satisfy this constraint, the client must give up the transmission opportunity if it wins the contention after t_{null}. The efficiency of the above protocol hence depends on the value of t_{null}. A large t_{null} defers the information exchange and the data packets of later contention winners, while a small t_{null} decreases the opportunity of concurrent transmissions. We will verify in §6.4 that setting $t_{null} = CW_{min}/2$ balances out the above tradeoff and produces a relatively low overhead.

---

[3]As we describe in §3.5, other contending clients use this preamble to count the number of ongoing transmissions.

Finally, we perform the following optimizations to minimize the overhead of the channel state information.

- Since 802.11 typically operates on a 20MHz OFDM channel, each client has to learn the channels over the 48 occupied OFDM subcarriers. It is however a high overhead to broadcast the direction of each subcarrier. We observe that after transforming the channels across all the 64 subcarriers to the time domain, there are only few taps with a noticeable amplitude. In particular, we empirically measure the amplitude of time-domain taps in the OFDM FFT window in our testbed. The result we plot in Fig. 8 shows that only about 5 taps have a relatively large amplitude. This is expected because there are only a limited number of distinguishable paths between a transmitter and a receiver in an indoor environment. Thus, a TurboRate client only announces the first few significant taps, e.g., 5 taps, of the time-domain channels. We will demonstrate in §6.1 that discarding the other taps results in a negligible error. The other clients can recover the channel information by transforming them back to the frequency-domain channels.

- The channel of each subcarrier in an M-antenna AP scenario is an M-dimensional vector, in which each element is the channel between the client's antenna to one of the AP's M antennas. Instead of sending the M-dimensional channel vector $\mathbf{h} = [h_1, h_2, \cdots, h_M]$, the client only requires to inform the direction of that vector, which is equivalent to the direction of a scaled vector $\mathbf{h}^d = [1, h_2/h_1, \cdots, h_M/h_1]$. Scaling the vector reduces the size of representing the direction to $M - 1$ complex numbers for each subcarrier.

After the above optimizations, the overhead of transmitting the direction information is 5 and 10 complex numbers for the 2-antenna and 3-antenna AP scenarios respectively, which are about 3 and 6 BPSK symbols.

3.3 Estimating the Best Bit Rate

Next we focus on how each client uses its SNR and the channel directions of ongoing transmissions to select its best bit rate. Lets consider a general scenario where a client wins the $(k + 1)^{th}$ contention and transmits a concurrent stream to an M-antenna AP in the presence of k ongoing transmissions. Let $\mathbf{h}_{k+1}$ denote the vector of the client's channels to the AP and $\mathbf{h}_i^d$, $i = 1, \cdots, k$, denote the directions, i.e., scaled channel vectors, of the k ongoing transmissions. To estimate its SNR after projection, SNR_{proj}, the client first estimates its own SNR to the AP, SNR_{orig}. Then the client subtracts the SNR reduction caused by the projection, ΔSNR, which as explained in §2, can be estimated using $\mathbf{h}_{k+1}$ and $\mathbf{h}_i^d$. Note that the subspace spanned by the k ongoing transmissions is the same as that spanned by their directions. We can therefore use the directions of the ongoing signals to compute the SNR reduction.

After estimating the SNR after projection in each OFDM subcarrier, the client computes the effective SNR (ESNR) [17], which can then be mapped to the best bit rate. ESNR considers the impact of frequency selectivity across multiple OFDM subcarriers, and hence can more accurately predict the bit rate.

3.4 Decoding at the AP

A simple way for an M-antenna AP to decode M concurrent streams is to use zero forcing for each of the streams. In particular, the AP decodes each stream by projecting the signal along the direction orthogonal to all the other concurrent streams. This decoder however might make the bit rates selected by the clients undecodable. To see why this is the case, lets consider a 3-antenna AP scenario where three clients communicate with the AP and join the concurrent transmissions one after another. Say the AP is interested in decoding the second stream. Recall that the second client estimates its SNR after projection, SNR_{proj}, according to the angle between its signal and the direction of the first client. If the AP ignores this fact and simply decodes the second stream by projecting along the direction orthogonal to both the first and the third clients, it will produce a SNR different from SNR_{proj}. This is because it projects on the orthogonal direction of a different subspace and leads to a different amount of SNR reduction after projection. Thus, the AP is unlikely to successfully decode the second client.

To ensure that the rate selected by each client can be decoded correctly, a TurboRate AP instead uses *zero-forcing with successive interference cancellation* (ZF-SIC) [33]. Using ZF-SIC, the AP decodes the kth stream after removing all the interfering streams that join after the kth stream. For example, lets again consider the 3-antenna AP scenario. The AP first decodes the third stream by projecting along the direction orthogonal to the plane of the first and the second clients. It then re-encodes the third stream and subtracts it from the received signals. The AP then decodes the second stream by projecting the resulting signal along the direction orthogonal to the first client. It then subtracts the second client and decodes the first client using a standard decoder.

Using ZF-SIC, the AP decodes the kth client after cancelling the interfering clients that joined after it. This property allows the AP to decode the packet sent at a rate chosen by the kth client, which the client computes using only its SNR and the channel directions of the $k - 1$ clients that won contention before it.

3.5 TurboRate's Medium Access Protocol

Fig. 7 shows an instantiation of our MAC in a network where the AP has three antennas and can support three concurrent transmissions. Similar to the MAC protocol in SAM [31], each client listens to the medium and counts the number of concurrent streams by cross-correlating with the known preamble. If the number of existing streams is less than the number of antennas supported by the AP, the clients contend for the medium using 802.11's contention window and random backoff. Clients can continue contending for transmission opportunities until they detect that the number of concurrent streams equals the number of antennas at the AP. Unlike the MAC protocol in SAM, TurboRate allows clients to contend for concurrent transmissions, only if their SNR after projection is higher than the 802.11 operational SNR range, i.e., 4 dB. In addition, the contention winner selects its best rate before data exchange, and annotates the direction of its channels in the header. To ensure that the clients can overhear the information annotated by the contention winners, TurboRate makes the ongoing transmissions pause their streams as mentioned in §3.2.

3.6 Supporturng Clients with Multiple Antennas

TurboRate's design can be easily generalized to support clients with multiple antennas. For example, a 2-antenna client and a single antennas client can transmit concurrently to a 3-antenna AP. TurboRate can be generalized to such scenarios by having a multi-antenna client contend for each of its antenna independently. The only difference, however, is that because radios typically cannot transmit and receive at the same time, a multi-antenna client contends for all the antennas it can transmit on, before starting the concurrent data transmission.

4. ADDITIONAL ISSUES

This section addresses the following additional issues.

Acknowledgements: Since the AP has multiple antennas, it can send the acknowledgements to all the clients concurrently on the downlink using beamforming [7].

Fragmentation and Aggregation: In TurboRate, all concurrent clients end their transmissions at about the same time. To do so, nodes may need to fragment or aggregate packets. TurboRate leverages the methods used in existing link layer protocols, e.g., packet fragmentation [18] and packet aggregation [6].

Retransmissions: A TurboRate client needs to re-transmit the packet if it is not ack-ed. The retransmission can be with a different subset of clients. The client thus must select a different rate and fragment or aggregate the packet differently.

Time Synchronization: To avoid inter-symbol interference (ISI), the concurrent clients need to synchronize their transmissions within a cyclic prefix of an OFDM symbol [31]. TurboRate applies the method proposed in [31], which allows concurrent clients to estimate the OFDM symbol boundary of the first stream and synchronize their transmission to it. To cope with the small delays due to hardware turn-around time and channel propagation, both the cyclic prefix and the OFDM FFT window are scaled up by the same factor. Such scaling does not increase the overhead, but allows the system to tolerate synchronization error [30].

Frequency Offset: To avoid inter-carrier interference, concurrent clients should have the same carrier frequency offset (CFO) at the AP. In TurboRate, all clients compensate their offset using a mechanism proposed in [28, 30]. Specifically, all the clients overhear the PLCP header sent by the first contention winner and estimate the frequency offset with respect to the first client. All the concurrent clients synchronize their frequency-domain signals by compensating this offset.

Fairness: In TurboRate, the first client that wins the contention for a concurrent transmission is likely to have a higher rate than the others since it computes its rate using its original SNR without projection. TurboRate however is still fair because every client wins the first contention with an equal probability, as in 802.11. In TurboRate, a client has the opportunity to transmit the first stream without lowering its rate. It can transmit concurrently and benefit from the throughput gain of MU-MIMO if it loses the first contention.

Imperfect Cancellation: In practice, Successive Interference Cancellation (SIC) cannot cancel the interfering signals perfectly, resulting in some residual noise. The magnitude of the residual noise depends on the power at which the interfering signal is received. A high residual noise might hinder the AP from decoding the existing signals after SIC. To avoid this, a TurboRate client determines whether it can join ongoing transmissions by applying a technique similar to the one proposed in [23]. In particular, a TurboRate client is only allowed to transmit concurrently if its interfering power at the AP is below a threshold.[4] This limits the amount of residual noise and allows the AP to successfully decode the existing signals after SIC.

Hidden Terminals: TurboRate provides a rate adaptation algorithm for MU-MIMO and is orthogonal to the issue of hidden terminals. In general, MU-MIMO enables concurrent transmissions and hence may increase the probability of hidden terminals. However, one can address the hidden terminal problem in MU-MIMO networks by using mechanisms like retransmissions [21], RTS-CTS [23, 22], and ZigZag [14].

---

[4]This threshold in practice is set to about 25–27dB [23].
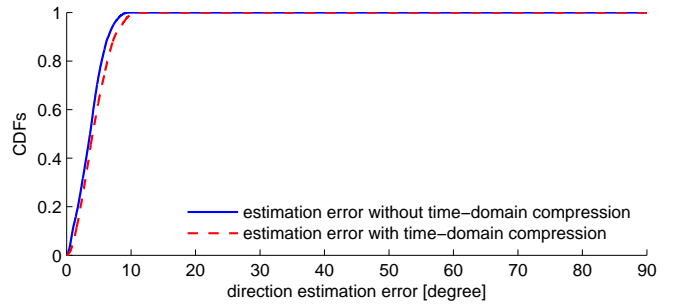
Figure 9—Accuracy of direction estimation. Leveraging channel reciprocity allows clients to measure the direction of the channels accurately in a passive way. The additional estimation error caused by compressing the time-domain channels is negligible.
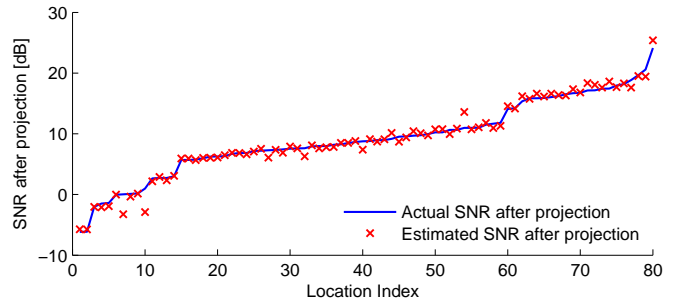
Figure 10—Accuracy of SNR estimation. The estimated SNR after projection closely matches the actual SNR after projection in the 802.11 operational SNR range.
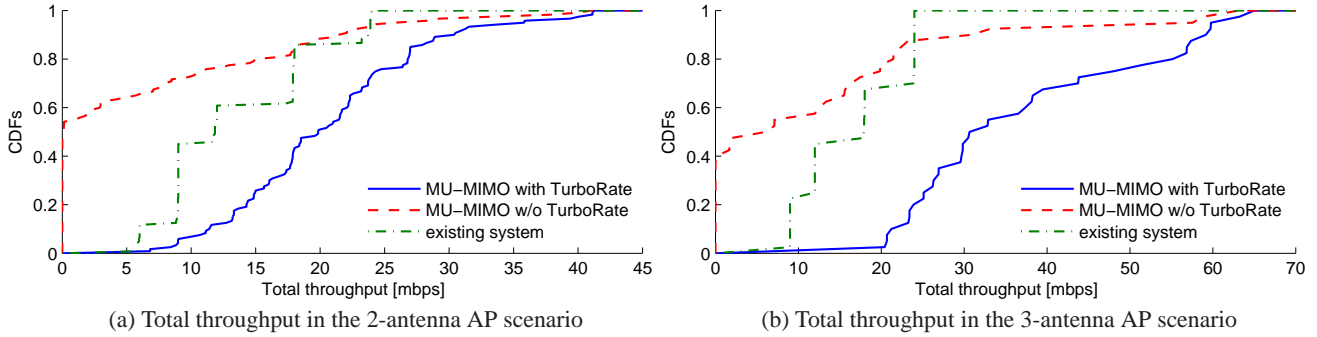
5. IMPLEMENTATION

We build a prototype of TurboRate using the USRP-N200 radio platform [2] and the UHD software package. Each USRP-N200 is equipped with an RFX2400 daughterboard, and operates on a 10 MHz channel. We build a multi-antenna AP by combining multiple USRP-N200 boards using an external clock [3] and making them act as a MIMO node. Each node runs a PHY layer similar to that in 802.11a, i.e., including OFDM subcarriers and using modulations (BPSK, 4-64QAM) and standard 802.11 code rates [5]. Since we operate at the bandwidth of 10MHz, the possible bit rates range from 3 to 27 Mb/s.

Due to the timing constraints limited by software radio, we implement all the components of our design except contention and ACK. To allow multiple clients to transmit concurrently, we leverage USRP-N200 timestamps to synchronize the clients within a cyclic prefix as follows. We make the AP broadcast a trigger signal. Each client records the timestamp of detecting the trigger, $t_{trigger}$, waits a pre-defined period of time, t_Δ, and sets the timestamp of the beginning of its transmission to $t_{start} = t_{trigger} + t_\Delta$. In our testbed, t_Δ is set to 0.1s, which is long enough to tackle the delays introduced by software.

6. RESULTS

We evaluate the performance of TurboRate in the testbed environment shown in Fig. 4. Our evaluation focuses on answering the following questions:

- Are the estimate of the direction of the channels and the SNR after projection accurate enough for a client to select its best bit rate?

(a) Total throughput in the 2-antenna AP scenario

(b) Total throughput in the 3-antenna AP scenario

Figure 11—Throughput Comparison. The figure compares the throughput with and without TurboRate for the 2-antenna and 3-antenna AP scenarios. TurboRate delivers a total throughput gain of 1.7x and 2.3x as compared to existing 802.11. Without considering the effect of projection, the concurrent client in many cases selects a rate that makes its stream undecodable after projection and leads to zero throughput.

- What is the throughput gain achieved by TurboRate?
- Where does the throughput gain come from?
- How much extra overhead is introduced by TurboRate?

6.1   Micro Benchmark

The performance of bit rate selection in TurboRate relies on the accuracy of two estimates: the directions of the concurrent clients, which the client learns from the annotation in their packets, and the client's computation of its SNR after projection. We empirically measure the accuracy of these two variables.

(a) Accuracy of Signal Direction Estimate: The errors of the signal direction estimate come from two sources: 1) the estimation error due to learning the channels using reciprocity, and 2) the information loss due to compression, i.e., due to expressing the channel using only 5 time-domain taps, as mentioned in §3.2. We check how these two errors impact the accuracy of the estimate.

Experiment: We consider a 2-antenna AP scenario where a single-antenna client communicates with the AP. The client and the AP are randomly assigned to the locations in Fig. 4. We measure their uplink and downlink channels, and calibrate the tx and rx chains using the method proposed in [4]. Since our protocol makes each client send only five taps to reduce the overhead, we further compare the accuracy of the direction after performing the following compression: convert the direction of the downlink channels after calibration to the time domain, keep only 5 taps and reset the rest to zero, and convert them back to the frequency domain. We define the estimation error as the angle between the direction of the actual uplink channel and the estimated direction of the channel, i.e., the direction of the downlink channel after calibration, with and without compression.

Results: Fig. 9 plots the CDFs of the estimation error across all experiments. The figure shows that the medium estimation error is only 4 degree, which means that the estimated direction is close to the actual direction. The additional estimation error caused by compression in time-domain information is negligible. The results show that clients can exploit the channel reciprocity property to estimate this information accurately in a passive way. Exchanging only 5 taps of time-domain information introduces a minimum estimation error, but decreases the overhead significantly.

(b) Accuracy of SNR Estimation: We next check how accurate a client can estimate SNR_{proj} using the method mentioned in §3.3.

Experiment: We focus on the scenario where two single-antenna clients communicate with a 2-antenna AP. In each experiment, the AP transmits 10 known symbols for the clients to learn their

channels using reciprocity, followed by both clients transmitting a 1500B data packet concurrently. We compress the direction of the channel as mentioned in the previous experiment, and use the channels learned from the known symbols and the noise at the AP to estimate the SNR after projection. We compare the estimated SNR after projection to the actual projected SNR, which is computed *at the AP* by using ZF to decode the received concurrent packets. We repeat the same experiment with different random locations of nodes in Fig. 4.

Results: Fig. 10 compares the estimated SNR to the actual SNR after decoding. The results show that the estimation is accurate when the SNR after projection is larger the 802.11 operational SNR, i.e., 4 dB, as shown in Fig. 10 along the y-axis. We however note that the estimation error in the extremely low SNR regime (i.e., lower than 4 dB) does not harm our system because OFDM does not work properly in this critical regime, and hence we do not allow the client to transmit concurrently. For the operational SNR regime, the average estimation error is about 0.5 dB, which has little impact on bit-rate selection.

6.2   Throughput Gain of TurboRate

We next investigate the throughput gain delivered by enabling TurboRate in MU-MIMO. We compare the throughput of three systems: 1) MU-MIMO with TurboRate, which is our proposed protocol, 2) MU-MIMO without TurboRate, in which clients transmit concurrently, but select their rates only according to their own SNRs to the AP without considering the interaction between the concurrent transmissions, and 3) the existing system, in which only a single client is allowed to transmit to a multi-antenna AP using diversity gain [33]. We compare their performance in 2-antenna AP and 3-antenna AP scenarios respectively.

Experiment: We first focus on the scenario in Fig. 1(a), where two single-antenna clients transmit concurrently to a 2-antenna AP. We repeat the experiment with random assignment of node locations in Fig. 4. Each experiment consists of three phases: First, two clients transmit concurrently at the rates selected by TurboRate. Second, both clients transmit concurrently at the bit rates selected based on their own SNRs to the AP without projection. Third, one of the two clients is picked randomly and made to transmit alone at the best bit rate supported by its own SNR without projection. In each phase, each concurrent client transmits a 1500 byte packet, and uses the ESNR [17] to lookup the optimal rate.

Results: Fig. 11(a) plots the CDFs of the total throughput of three different systems. The figure shows that enabling TurboRate in

(a) Throughput of the client decoded by ZF (b) Throughput of the client decoded by SIC
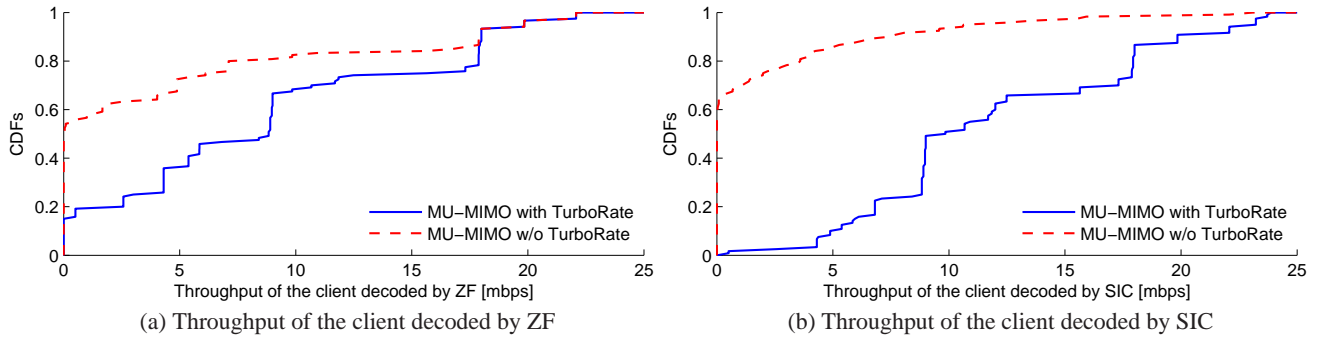
Figure 12—Throughput of individual clients. Without TurboRate, the client has a high probability to pick a wrong rate, which is undecodable by zero forcing and leads to zero throughput. With TurboRate, it can not only select the optimal rate, but also determine whether it should refrain from transmitting concurrently due to an extremely low SNR after projection. This is why the other client can be decoded correctly after interference cancellation and still achieve high throughput in TurboRate.

a MU-MIMO network ensures decodability and allows clients to achieve high throughput. Compared to existing 802.11 where only one client is allowed to transmit, the throughput gain from enabling concurrent transmissions with TurboRate's bit rate selection is about 1.7x, matching the analysis in §2. In contrast, concurrent transmissions with traditional bit rate adaptation could cause one client to be decoded incorrectly and leave residual interference, as a result harming the other client. The results show that traditional bit rate selection hampers the gain of MU-MIMO, and leads to large throughput reductions compared to existing 802.11 (about 50% of the cases are reduced to zero throughput).

Experiment: We next check the performance in a 3-antenna AP scenario where three single-antenna clients transmit concurrently to the AP. Each experiment consists of three phases: In phases 1 and 2, three clients transmit 1500 byte packets concurrently at the bit rate selected by TurboRate and selected only based on their own SNRs, respectively. In phase 3, we randomly pick one of the three clients and make it transmit alone at its best rate. We repeat the experiment with random assignment of nodes locations in Fig 4.

Results: Fig. 11(b) plots the CDFs of the total throughput of the three compared systems. The total network throughput of Turbo-Rate in the 3-antenna AP scenario increases by 2.3x over existing 802.11 where only one stream is allowed. Further, without Turbo-Rate, the gain of MU-MIMO cannot be achieved. Note that the throughput of three concurrent MU-MIMO clients is *not* three times as high as a single client. The reason is that the second and third concurrent clients lose some of their SNRs due to projection. This is a natural limitation of MU-MIMO (not a limitation of Turbo-Rate.)

6.3 Implications of Not Using MU-MIMO Rate Adaptation

To better understand TurboRate's throughput gains, we zoom in on the throughput that the individual clients can achieve in the 2-antenna AP experiment mentioned in the last section. TurboRate decodes one client using zero-forcing (ZF), projecting the received signal along the direction orthogonal to the other client. It decodes the other client using successive interference cancellation (SIC), i.e., it is decoded after removing the interfering signal decoded by ZF.

Results: We first plot in Fig. 12(a) the throughput of the client decoded by ZF. Our findings are:

- Without considering the effect of projection, the client is very likely to choose a bit rate that exceeds its capacity after projection. This results in 54% of experiments with zero throughput.
- For 15% of the experiments, the SNR after projection is lower than the 802.11 operational SNR range. Most of these cases occur when the client is in the low original SNR regime and thus more sensitive to SNR reduction after projection. For this critical regime, TurboRate plays an important role to enable the client to detect these situations and prevent interfering with the ongoing transmission by refraining from transmitting concurrently.

We next plot in Fig. 12(b) the throughput of the client decoded by SIC. The figure shows the following:

- Without TurboRate, the AP cannot remove the interference from the other client because it did not adapt to SNR reduction after projection and picked a wrong rate. In this case, the AP cannot decode the other client and subtract its signal and hence also fails to decode this client correctly using SIC. This reduces the throughput of this client to zero as well. The client decoded by SIC can only obtain positive throughput if the angle between the two clients is by chance large enough such that the AP can still decode the other client correctly even after projection.
- In TurboRate, the client decoded by SIC can achieve a throughput comparable to that when it transmits alone because the AP can correctly decode and remove the interfering client. There might be a small residual interference left after interference cancellation due to imperfect hardware linearity. The results however show that this small interference does not hinder the AP from decoding the client after interference cancellation.

The above results verify that TurboRate is not only necessary for the client that joins the ongoing transmission and is decoded by projection, but also essential for the client that wins the earlier contention.

6.4 Overhead

We finally check how much extra overhead is introduced by TurboRate due to direction announcement. The overhead includes two parts: 1) the transmission time required for sending the annotated information, and 2) the idle period for ensuring correct reception of the information. To analyze the overhead, we have to consider the dynamics of node contention in a large scale network. This is hard to do in a USRP testbed because of the long delay and the difficulty of experimenting with realtime contention. We hence use Matlab to simulate the dynamics of 802.11 contention
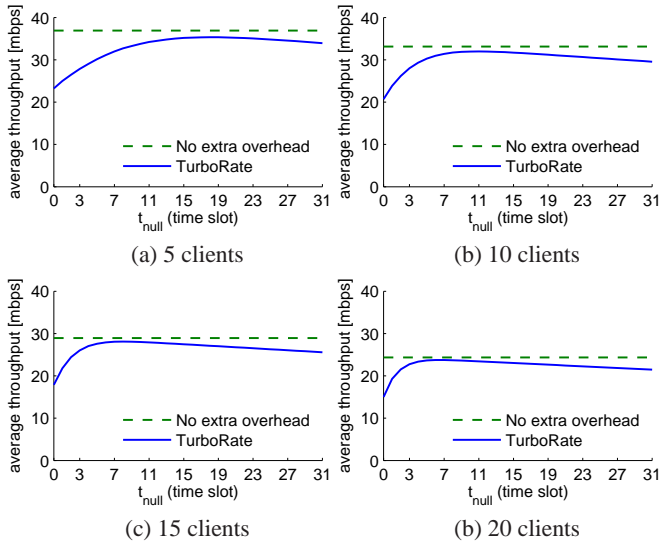
Figure 13—Overhead in a 3-antenna AP scenario. By setting $t_{null} = CW_{min}/2$, the extra overhead introduced by TurboRate results in 4% throughput reduction, which is fairly small.

in a large network. We use a 3-antenna AP scenario with many single-antenna clients that contend for concurrent transmissions.

In the simulations, we implement all the components of our protocol, including contention, backoff, interframe timing (i.e., SIFS and DIFS), PLCP header, MAC header and ACK. We randomly assign a channel vector to each client, and assume that all clients always have packets to send. The first contention winner transmits a 1500-byte packet, and the second and third winners end their transmissions at the same time as the first client. Each simulation compares the average total throughput of 10,000 transmissions [5] of the following schemes: 1) no extra overhead: each client starts sending its data packet immediately after it wins the contention without any overhead of information exchange, and 2) TurboRate: the first client pauses its transmission at a pre-defined time-slot t_{null} such that the second client can broadcast its direction to the rest of clients. We test different values of t_{null} and force all clients to give up the opportunity of the second transmission if no one rolls a random number smaller than t_{null} in the second contention. Clients can however start the third contention after t_{null}, regardless of the outcome of the second contention. This is because the AP has only 3 antennas and hence the third client does not need to announce its direction to other potential contenders.

Results: Fig. 13 plots the throughput of two schemes for varying numbers of clients. The figures show that the throughput decreases with increasing number of clients due to increased probability of collisions. A small t_{null} decreases the concurrent transmission opportunity, while a large t_{null} forces the client who wins the third contention to wait for the information and postpone its transmission. The maximum throughput can be achieved by balancing the above tradeoff and picking the optimum t_{null}. The optimum choice of t_{null} however changes with the number of contending clients. To avoid the complexity, we can simply set $t_{null} = 15$, i.e., half of CW_{min} defined in 802.11. After that, as compared to the throughput without extra overhead, the average throughput loss due to TurboRate's

---

[5]We compute the throughput as the total number of bits in the correctly-decoded packets divided by the channel time occupied by 10,000 transmissions, where each transmission includes up to three concurrent packets.

overhead, including the loss of concurrent transmission opportunities and the time used for exchanging information, is 4%, which is fairly small.

7. RELATED WORK

Related work falls in the following two areas:

(a) Multi-user MIMO WLANs: MU-MIMO advocates having multiple clients concurrently communicate with a single AP or multiple receivers. The gain of MU-MIMO WLANs has been verified theoretically [11, 12, 34] and realized empirically [31, 7, 37, 15, 23]. SAM [31] allows multiple single-antenna clients to communicate concurrently with a multi-antenna AP. Beamforming [7, 37] deals with the downlink, and allows an AP to communicate concurrently with multiple single-antenna clients. IAC [15] makes multiple APs connect to each other and act as a virtual MIMO node to communicate with multiple clients concurrently. All these practical MU-MIMO systems leave rate adaptation an open issue. 802.11n[+] [23] enables random access for concurrent MIMO transmissions and takes bit rate selection into account. 802.11n[+] however works only when nodes have different numbers of antennas. It also uses an handshake to exchange information which introduces additional overhead. In contrast, TurboRate enables distributed bit rate adaptation for MU-MIMO LANs without explicit coordination and can support clients, with the same or different numbers of antennas, to communicate concurrently with an AP.

(a) Bit rate adaptation: There is a rich literature on rate adaptation for legacy 802.11a/b/g. They assume that the channels do not change for a short period of time, and hence can exploit historical performance, like loss rate [8, 36], SNR [17, 27, 9], BER [24], soft values [35], to predict the optimal bit rate for the next packet. Historical-based rate adaptation is then extended to 802.11n MIMO networks, where two multi-antenna nodes communicate with each other [26, 25, 10]. Such assumption holds for a single pair of MIMO nodes, but not in MU-MIMO, in which the best bit rate of a client changes with its concurrent transmitters. Prior work on downlink MU-MIMO rate adaptation [20, 32] enables a single transmitter to select the best rates for all its clients. Uplink rate adaptation is however much more difficult because, in a random access network, a client could transmit concurrently with a different subset of clients and hence requires to adapt its rate on a per-packet basis depending on who are the current transmitters. Compared to prior uplink MU-MIMO rate selection algorithms [38, 19], which require the AP to explicitly coordinate between the clients for each packet and tell each client the bit rate it has to use, TurboRate enables distributed rate adaptation and hence can be applied in a random access network.

8. CONCLUSION

This paper introduces TurboRate, a distributed rate adaptation protocol for MU-MIMO LANs. It decomposes rate adaptation in dynamic MU-MIMO LANs to the estimation of two variables: the SNR when a client transmits alone to the AP and the direction of the client's signal received at the AP. The short-term stability of these two parameters allows each TurboRate client to measure them in a passive way, but still be able to adapt its optimal rate on a per-packet basis, depending on who are the concurrent clients. Our prototype implementation shows that enabling MU-MIMO with traditional rate adaptation reduces the throughput in most cases, while enabling TurboRate in MU-MIMO increases the network throughput by 1.7x and 2.3x over existing 802.11 for 2- and 3-antenna AP scenarios respectively.

9. ACKNOWLEDGMENTS

10. REFERENCES

[1] Cisco Aironet 3600 Series. http://www.cisco.com/en/US/products/ps11983/index.html.

[2] Ettus Inc., Universal Software Radio Peripheral. http://ettus.com.

[3] Jackson Labs Technologies Inc., Fury GPSDO. http://www.jackson-labs.com/index.php/products/fury.

[4] System Description and Operating Principles for High Throughput Enhancements to 802.11. IEEE 802.11-04/0870r, 2004.

[5] Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11*, 2007.

[6] *IEEE Std 802.11n-2009*, pages c1 –502, 2009.

[7] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly. Design and Experimental Evaluation of Multi-User Beamforming in Wireless LANs. In *ACM MobiCom*, 2010.

[8] J. Bicket. *Bit-rate Selection in Wireless Networks*. PhD thesis, Massachusetts Institute of Technology, 2005.

[9] J. Camp and E. Knightly. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-layer Implementation and Experimental Evaluation. In *ACM MobiCom*, 2008.

[10] C.-B. Chae, A. Forenza, R. Heath, M. McKay, and I. Collings. Adaptive MIMO Transmission Techniques for Broadband Wireless Communication Systems. *IEEE Communications Magazine*, 48(5):112 –118, may 2010.

[11] A. El Gamal and T. Cover. Multiple User Information Theory. *Proceedings of the IEEE*, 68(12):1466 – 1483, dec. 1980.

[12] D. Gesbert, M. Kountouris, R. Heath, C.-B. Chae, and T. Salzer. Shifting the MIMO Paradigm. *IEEE Signal Processing Magazine*, 24(5):36 –46, sept. 2007.

[13] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the RF Smog: Making 802.11n Robust to Cross-technology Interference. In *ACM SIGCOMM*, 2011.

[14] S. Gollakota and D. Katabi. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *ACM SIGCOMM*, 2008.

[15] S. Gollakota, S. D. Perli, and D. Katabi. Interference Alignment and Cancellation. In *ACM SIGCOMM*, 2009.

[16] M. Guillaud, D. Slock, and R. Knopp. A Practical Method for Wireless Channel Reciprocity Exploitation through Relative Calibration. In *Signal Processing and Its Applications*, 2005.

[17] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. In *ACM SIGCOMM*, 2010.

[18] R. Handel and M. Huber. *Integrated Broadband Networks; An Introduction to ATM-Based Networks*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[19] Y. Kim, S. Cho, and D. K. Kim. Low Complexity Antenna Selection based MIMO Scheduling Algorithms for Uplink Multiuser MIMO/FDD System. In *IEEE VTC Spring*, 2007.

[20] V. Lau and M. Jiang. Downlink Scheduling and Rate Adaptation Design of Multi-User, Multiple-Antenna Base Station with Imperfect CSIT. In *IEEE GLOBECOM*, 2005.

[21] L. E. Li, K. Tan, H. Viswanathan, Y. Xu, and Y. R. Yang. Retransmission ≠ Repeat: Simple Retransmission Permutation Can Resolve Overlapping Channel Collisions. In *ACM MOBICOM*, 2010.

[22] K. C.-J. Lin, Y.-J. Chuang, and D. Katabi. A Light-Weight Wireless Handshake. *ACM SIGCOMM Comput. Commun. Rev.*, 42(2):28–34.

[23] K. C.-J. Lin, S. Gollakota, and D. Katabi. Random Access Heterogeneous MIMO Networks. In *ACM SIGCOMM*, 2011.

[24] K. C.-J. Lin, N. Kushman, and D. Katabi. ZipTx: Harnessing Partial Packets in 802.11 Networks. In *ACM MobiCom*, 2008.

[25] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu. MIMO Rate Adaptation in 802.11n Wireless Networks. In *ACM MobiCom*, 2010.

[26] F. Peng, J. Zhang, and W. Ryan. Adaptive Modulation and Coding for IEEE 802.11n. In *IEEE WCNC*, 2007.

[27] H. Rahul, F. Edalat, D. Katabi, and C. Sodini. Frequency-Aware Rate Adaptation and MAC Protocols. In *ACM MOBICOM*, 2009.

[28] H. Rahul, H. Hassanieh, and D. Katabi. SourceSync: a Distributed Wireless Architecture for Exploiting Sender Diversity. In *ACM SIGCOMM*, 2010.

[29] G. Ren, H. Zhang, and Y. Chang. SNR Estimation Algorithm based on the Preamble for OFDM Systems in Frequency Selective Channels. *IEEE Transactions on Communications*, 57(8):2230 –2234, aug. 2009.

[30] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang. Fine-Grained Channel Access in Wireless LAN. In *ACM SIGCOMM*, 2010.

[31] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. M. Voelker. SAM: Enabling Practical Spatial Multiple Access in Wireless LAN. In *ACM MobiCom*, 2009.

[32] M. Torabi, W. Ajib, and D. Haccoun. Discrete-Rate Adaptive Multiuser Scheduling for MIMO-OFDM Systems. In *IEEE VTC 2008-Fall*, 2008.

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

[34] P. Viswanath and D. Tse. Sum Capacity of The Vector Gaussian Broadcast Channel and Uplink-downlink Duality. *IEEE Transactions on Information Theory*, 49(8):1912 – 1921, aug. 2003.

[35] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-Layer Wireless Bit Rate Adaptation. In *ACM SIGCOMM*, 2009.

[36] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *ACM MobiCom*, 2006.

[37] H. Yu, L. Zhong, A. Sabharwal, and D. Kao. Beamforming on Mobile Devices: a First Study. In *ACM MobiCom*, 2011.

[38] Y. Zhang, C. Ji, Y. Liu, W. Malik, D. O'Brien, and D. Edwards. A Low Complexity User Scheduling Algorithm for Uplink Multiuser MIMO Systems. *IEEE Transactions on Wireless Communications*, 7(7):2486 –2491, july 2008.