Multi-point to Multi-point MIMO in Wireless LANs

Sangki Yun, Lili Qiu, Apurv Bhartia
The University of Texas at Austin

***Abstract*—Distributed multiple-input multiple-output (MIMO) promises a dramatic capacity increase. While significant theoretical work has been done on distributed MIMO at the physical layer, how to translate the physical layer innovation into tangible benefits to real networks remains open. In particular, realizing multi-point to multi-point MIMO involves the following challenges: (i) how to accurately synchronize multiple APs in phase and time in order to successfully deliver precoded signals to the clients, and (ii) how to develop a MAC protocol to effectively support multi-point to multi-point MIMO. In this paper, we develop a practical approach to address the above challenges. We implement multi-point to multi-point MIMO for both uplink and downlink to enable multiple APs to simultaneously communicate with multiple clients. We examine a number of important MAC design issues, such as how to access the medium, perform rate adaptation, support acknowledgments in unicast traffic, deal with losses/collisions, and schedule transmissions. We demonstrate its feasibility and effectiveness through a prototype implementation on USRP and SORA, two of the most well-known software defined radio platforms.**

I. Introduction

Motivation: Multiple-input multiple-output (MIMO) is an exciting technology that promises a dramatic capacity gain by exploiting spatial multiplexing and diversity across multiple transmitter and receiver antennas. While significant research has been done on the theoretical front, realizing multi-point to multi-point MIMO in practice requires addressing several significant challenges. First, antennas residing on different nodes are driven by different clocks, so it is challenging to synchronize the phase and time of their transmissions. Second, the new style of transmissions requires us to revisit the MAC design, such as how to access the medium, adapt rates, support ACKs, handle losses and collisions, schedule transmissions, limit Ethernet overhead, and obtain channel state information.

Our approach: In this paper, we develop a practical approach to enable distributed MIMO in WLANs as follow.

- For the downlink, we implement Zero-Force (ZF) beamforming to allow multiple APs to transmit to multiple clients simultaneously. The APs perform joint precoding such that the combined signals arriving at each client can be demodulated as usual. The major challenge in realizing this goal is to precisely synchronize phase and time of their transmissions. We develop a practical solution to address these challenges (Section III).
- For the uplink, we let multiple APs share their received signals and perform joint decoding of multiple simultaneous uplink transmissions (Section IV).
- We address a number of important MAC issues, including how to access the medium, perform rate adaptation, support acknowledgments in unicast traffic, deal with losses/collisions, and schedule transmissions (Section V).
- We implement our approach on USRP [13] and SORA [10], two of the most well known software-defined radio platforms to demonstrate its feasibility and effectiveness (Section VI).

Our work advances the state-of-art on point-to-point MIMO (*e.g.*, IEEE 802.11n) and point-to-multi-point MIMO (*e.g.*, IEEE 802.11ac) in that when each AP has multiple antennas we can now send up to the sum of the total numbers of antennas across all APs. For example, using our approach, three 8-antenna APs can support up to 24 concurrent streams instead of limiting to 8 streams as in 802.11ac. Moreover, we go beyond the previous works (*e.g.*, [1], [11], [8], [2]), which all focus on the physical layer, by developing multi-point to multi-point MIMO-aware MAC.

II. Overview

Downlink: To illustrate the idea, let us start with a simple scenario with two APs sending to two clients, where two APs are connected using the Ethernet and are close enough to interfere with each other in the wireless medium. Consider we have two frames p_1 and p_2 to transmit to client 1 and 2, respectively. In traditional transmission, to avoid interference, only one of the APs can transmit at a time if they are on the same channel. So it takes two time slots to transmit p_1 and p_2.

Zero-Force (ZF) multi-user beamforming [12] can be used to deliver these two frames simultaneously. It makes the combined received signals arriving at each client the same as the clean intended signal for that client. Let H be a channel coefficient matrix, where its element at the i-th row and j-th column h_{ij} is the channel coefficient from AP i to client j. If a transmitted signal is a product of the inverse of the channel coefficient matrix and the intended signal vector, then the received signal at each client is the intended signal itself. To be more specific, the precoding is performed as follows:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{\|H^{-1}\|_F} H^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \|H\|_F H^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix},$$

where x_1 and x_2 are precoded signals transmitted by AP 1 and AP 2, p_1 and p_2 are the intended signal vectors, and $\|H^{-1}\|_F$ is the Frobenius norm of H^{-1}. We normalize by $\|H^{-1}\|_F$ to ensure Multi-point to Multi-point MIMO has the same power consumption as separate transmissions. The received signals are $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = H \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = H\|H\|_F H^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \|H\|_F \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$. Since each client receives its intended signal (just scaled down by the channel attenuation), it can decode the signal using traditional demodulation. This precoding method can be easily extended to more than 2 sender-receiver pairs. When n APs send to n clients, H is simply an $n \times n$ channel coefficient matrix, and we precode data using H^{-1}. When the number of APs is larger than the number of clients, the precoding matrix $\mathbf{W}$ is pseudo-inverse of the channel matrix to further achieve the diversity gain, which is $\mathbf{W} = \mathbf{H^T}(\mathbf{HH^T})^{-1}$. As before, $\|\mathbf{W}\|_F$ is used for normalization to ensure the same transmission power as separate transmissions. While we focus on ZF beamforming, our system design and synchronization

techniques can be used to support other precoding algorithms (*e.g.*, Dirty Paper Coding).

This precoding can also easily take advantage of multiple antennas on the same APs since the precoding procedure is the same regardless whether the antennas belong to the same APs or different APs. As a result, it can support $min(\sum_i N(i), \sum_j M(j))$ concurrent downlink streams, where $N(i)$ is the number of antennas at AP i and $M(j)$ is the number of antennas at client j. This is a fundamental advantage of multi-point to multi-point MIMO over point-to-point or point-to-multipoint MIMO, whose capacity is limited by the number of antennas residing on one AP. In comparison, Multi-point to Multi-point MIMO allows us to scale WLAN capacity by deploying more APs, which is much easier and cheaper to do than getting more spectrum.

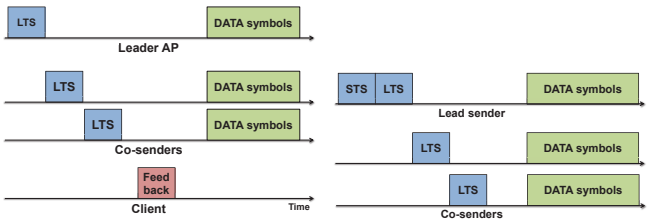
Uplink: In the uplink scenario, multiple clients simultaneously send traffic to APs and APs can cooperate over the Ethernet to decode the transmissions. For example, Let t_1 and t_2 denote the transmitted signals from client 1 and 2, respectively. The APs 1 and 2 receive $y_1 = h_{11}p_1 + h_{12}p_2$ and $y_2 = h_{21}p_1 + h_{22}p_2$, respectively. The APs can decode the unknown transmitted signal t_1 and t_2 by measuring and sharing the channel coefficients and received signal and solving a linear system with 2 unknowns based on the 2 constraints derived from the received signals. In this way, it can support $min(\sum_i N_i, \sum_j M_j)$ streams in the uplink of WLANs.

Architecture: To generalize the above downlink/uplink examples, we have a controller that coordinates all the APs. In the downlink case, the controller precodes the signals and transmits the precoded signals to the responsible APs, which will further transmit to the air. In the uplink, the controller gathers the received signals from all the involved APs and performs joint decoding. The Ethernet traffic required in both uplink and downlink is affordable.

III. Multiplexing Downlink Traffic

In this section, we describe how to multiplex downlink traffic using distributed and loosely synchronized APs.

Need to synchronize phase: In a standard point-to-point MIMO where multiple transceiver modules are attached in one hardware, the transceiver shares a single clock for Voltage Controlled Oscillator (VCO). In this case, all transmitted signals from different antennas have the same amount of Carrier Frequency Offset (CFO). In Multi-point to Multi-point MIMO, each AP has its own clocks, so their CFOs are different. This results in different initial phase offsets among transmitters, which cause serious problem in the Multi-point to Multi-point MIMO. When transmitters have different initial phases, the precoding introduced in Section II does not diagonalize the effective channel. In a case of 2 senders and 2 receivers, for example, $e^{j\Delta_1}$ and $e^{j\Delta_2}$ denote the initial phases of the two senders. The transmitted precoded signals are $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} e^{j\Delta_1} & 0 \\ 0 & e^{j\Delta_2} \end{bmatrix} H^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$. Then the received signals at the two receivers become: $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = H \begin{bmatrix} e^{j\Delta_1} & 0 \\ 0 & e^{j\Delta_2} \end{bmatrix} H^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}$. As a result, the effective channel is not diagonalized as intended due to the initial phase matrix as long as $\Delta_1 \neq \Delta_2$. In this case, each receiving signal is no longer just a function of its own signal but depends on the other signal, so it does not have enough information to correctly decode its signal.



(a) *Downlink phase synchronization* (b) *Uplink synchronization*

Fig. 1. Synchronization in the downlink and uplink.

Approaches to synchronize the phase: Synchronizing the initial phases among multiple transmitters is, however, feasible considering that the initial phase is a function of time and CFO. Let f_c be the difference between the center frequency and the hardware frequency at the sender (i.e., CFO) and $\angle e^{j\gamma_{t_d}}$ be the initial phase of the signal transmitted at time t_d, then the initial phase at time t is: $c(t) = \angle e^{j(\gamma_{t_d} 2\pi f_c (t-t_d))}$, where $\angle$ represents the phase of a complex number. This implies that the sender can predict the phase of its next transmission if it knows the time interval between transmissions, CFO, and the phase of the previous transmission. Therefore we propose a phase synchronization algorithm that lets all the transmitted signals have the same initial phase. Let t_i and $e^{j\gamma_i}$ be the transmission time and the initial phase of the ith sender's signal, respectively. Then the signal of the sender i at time t has the initial phase $c_i(t) = \angle e^{j(\gamma_i 2\pi f_c^i (t-t_i))}$. To synchronize the initial phases, all senders except the lead sender adjust their phases by multiplying the phase compensation factor $\alpha_i = e^{j(c_1(t) - c_i(t))}$ to all the transmitted signals so that all the transmitters have the same initial phase $c_1(t)$.

Now the only question remains is how to estimate CFO and the initial phase for the above computation. To achieve this, as shown in Figure 1(a), upon receiving the trigger message, the APs transmit preambles to estimate their initial phase and CFO. Starting from the lead AP, each AP transmits the preamble as ordered in the trigger frame. We use the Long Training Sequence (LTS) in IEEE 802.11a/g as preamble. To estimate the CFO, the receiver measures the amount of the phase rotation between two OFDM symbols in LTS using the algorithm in [5]. To further enhance the accuracy of CFO estimation, frequency-domain residual CFO estimation based on pilot symbols in [6] can be used. We handle sampling frequency offset during the packet by using a long-term averaged CFO estimate. To estimate the initial phase, the receiver measures the average phase difference between the known preamble and the received preamble using the same algorithm as in [5].

The measurement of CFO and initial phase can be done either by a client or by APs that are co-senders. To minimize the overhead, multiple data frames can be transmitted continuously after a single phase synchronization process.

Besides synchronizing the initial phase, we also need to synchronize the phase of all the signals involved in the transmissions. This is necessary because CFO difference across the APs can cause phase shift even though the initial phase is synchronized. To avoid this, each transmitter i except the lead sender multiplies $e^{j2\pi(f_c^1 - f_c^i)t_n}$ to its data signals, where t_n

is the time elapsed from the start time of the transmission to the time of transmitting the n-th signal so that all concurrent signals have the same phase.

IV. Multiplexing Uplink Traffic

Next we turn to multiplexing uplink traffic.

Synchronizing Transmissions: When clients have traffic to upload, they contend for the wireless medium using the traditional IEEE 802.11. Upon winning, a client may inform other clients to join its transmission using the same trigger message as described in Section III. To get the channel coefficient of each individual link, APs need a clean preamble from each transmitter. To do that, as shown in Figure 1(b), the lead sender transmits long training sequence (LTS) for channel estimation along with short training sequence (STS), and the kth co-sender transmits LTS after waiting $(k+1) \times |LTS|$. After the preamble transmission is finished, data symbols are transmitted. A client can select other clients to join its transmission by monitoring the wireless traffic and extracting other active clients' addresses. When a selected client has no traffic to send, it just skips its transmission and the receiver can detect the corresponding preamble is missing and properly decode the remaining signals.

Frame Decoding: The controller collects the received signals and estimates channel coefficients to decode the received frame using MIMO decoding schemes. One simple way is Zero Forcing (ZF), which multiplies the inverse of the channel coefficient matrix to the received signals. We use this in our prototype implementation for simplicity. The optimal method is Maximum Likelihood (ML) decoding, which finds the best possible combination of modulated symbols that closely matches the received signals. It incurs a high computation cost. Another alternative is Bell Laboratories Layered Space-Time (BLAST) [3] based on successive interference cancellation, which is sub-optimal but computationally more efficient than ML decoding method.

Handling Offset: One important difference between uplink and downlink multiplexing is that the uplink multiplexing does not require sample level timing synchronization, as the signals are not precoded. As long as the amount of timing offset is less than OFDM Cyclic Prefix (CP) duration, the receiver can properly decode it. To ensure that, we use a larger FFT window to proportionally increase the entire symbol time including CP. This does not increase CP overhead. In 802.11a WLAN based on 64-point FFT, a CP duration is 0.8 μs. To increase the slack time of synchronization without increasing the CP overhead, we use 256-point FFT, which gives 3.2 μs CP. The phase synchronization is not required in uplink transmission, because the phase offset is easily compensated in the channel compensation process, just as in a normal individual transmission (*i.e.*, a preamble from a given sender contains the same phase offset as that in its data signals and can be cancelled out).

However, CFO differences should be compensated in both uplink and downlink. To handle this, a client sends a sinusoid sequence every 10 seconds, and the other clients tune their center frequencies to reduce CFO. Based on our observation, CFO does not distort the signal as long as the CFO difference is maintained within 100Hz for a 20MHz channel. Note such a loose synchronization is not acceptable for downlink because the phase rotates even by a small CFO. Therefore we have to perform per-frame synchronization in the downlink case.

V. MAC Design

In this section, we examine a range of important MAC design issues: how to (i) access wireless medium, (ii) adapt data rates, (iii) support ACKs in unicast traffic, (iv) deal with losses and collisions, (v) schedule transmissions, and (vi) obtain channel state information.

Medium access: The controller selects a set of APs to transmit several frames simultaneously. The selected APs as well as clients use the IEEE 802.11 MAC protocol to contend for the medium as usual. Whenever one of the selected APs wins the medium, the winning AP sends a trigger frame to inform all the other APs to join its transmission. The trigger frame includes the number of co-senders, the co-sender addresses, and the corresponding client addresses. The trigger frame has network allocation vector (NAV) set to the duration from the end of the trigger message till the end of the data frame in order to prevent other nodes from interrupting the transmission.

When a client wins the medium, it sends a trigger message to inform other clients to join the transmission. As in the downlink, the NAV in the trigger is set appropriately to prevent other nodes from transmitting.

Rate adaptation: In the downlink case, a receiver's signal is generated by the composition of the signals of multiple transmitters. In this case, it is hard to estimate the SNR of the received signal by measuring the SNR of the individual link, which makes the rate selection problem challenging. To overcome this issue, unlike traditional RSSI-based SNR that divides the received signal strength by noise power, we use Error Vector Magnitude (EVM) of the received symbols to quantify the channel quality. EVM-based SNR measures the relative distance between the received symbols and the closest constellation points [9]. Therefore, we can estimate the signal quality that each receiver perceives without considering the signal strength of links. The same approach applies to the uplink traffic, where each AP estimates EVM according to the decoded symbols and their closest constellation points.

In addition, we should also take into account SNR difference across subcarriers as shown in [4]. To achieve the goal, the receiver estimates per subcarrier BER using the EVM and data rate, derives the Effective SNR (ESNR) that takes into account FEC according to [4], and feeds it back to the sender. Upon receiving it, the sender selects the rate that maximizes the throughput from SNR-BER relationship in [4]. Similarly, in the uplink, each AP computes EVM for each subcarrier and feeds the derived ESNR back to its sender to select the best data rate.

Supporting ACKs: So far we have focused on transmitting traffic in one direction. This is sufficient for broadcast and multicast traffic. For unicast traffic, we also need to support sending ACKs in the reverse direction. By using downlink and uplink transmission mechanism together, Multi-point to Multi-point MIMO can easily support ACKs. Specifically, when APs send unicast frames to clients, all the clients that receive the frames can simultaneously send ACKs back to APs in a similar way as sending multiple data frames to the APs (*i.e.*, they send preambles in clean and

then send ACKs together). The only difference is that we no longer need the trigger message. The clients can synchronize their ACK transmissions by all transmitting ACKs right after SIFS since data reception. The APs put the unacknowledged frames back to the global queue for retransmission. As initial transmissions, retransmissions can also be sent together with other frames in the future.

Similarly, when clients unicast data frames to APs, APs will precode ACKs in the same way as precoding data frames (*i.e.,* code the transmissions such that each client receives its intended ACK). Again, the APs can synchronize their ACK transmissions after SIFS since data reception. Clients retransmit if they do not receive their intended ACKs.

Dealing with Losses and Collisions: All transmissions on wireless are subject to losses and collisions. We leverage spatial diversity and rate adaptation to reduce losses of data frames and use ACKs and retransmissions to recover from data losses. So below we focus on losses of trigger messages. First, we consider losses due to weak wireless signal (but not due to interference). If a client misses a trigger message from the lead client, it is not harmful since the client that misses the trigger simply does not participate in the transmission and the APs can identify which clients actually transmit based on the headers that are transmitted and correctly decode the remaining transmissions. In contrast, if an AP misses a trigger message from the lead AP, it can cause decoding error since precoding ensures each client receives its intended signal only when all the APs involved transmit. To ensure the delivery of a trigger message, the lead AP sends a trigger message over both wireless and Ethernet. We cannot skip the trigger message over wireless because the NAV in the trigger message sent over wireless prevents other un-triggered transmitters from sending at the same time using virtual carrier sense.

Note that there is one special case where the triggered AP, denoted as AP A, senses the carrier and is not allowed to transmit at that time. This is possible because the APs are at different locations and may sense different carriers. In this case, the data transmissions without all the involved transmitters participating will fail. To minimize the overhead of failures, we let AP A send an abort message over the Ethernet. This not only informs all the other transmitting APs to stop wireless transmission immediately, but also informs all the other APs to cancel the NAV from the previous trigger message. In this way, the APs can proceed immediately to the next transmissions, thereby reducing the failure overhead.

Next we consider collision losses of trigger messages. When multiple trigger messages collide, multiple lead senders as well as those that receive the trigger message correctly (if any) proceed to send subsequent data frames, which result in data collision. If not dealt with, the cost of such collision is similar to data collision in the current IEEE 802.11 since the data frame typically dominates the transmission time. We can optionally reduce collisions by supporting RTS/CTS as follows. The trigger message from a lead AP serves as RTS. The clients that deem the medium is available will transmit CTS in the same way as sending ACKs. If the APs receive CTS correctly from all the clients through joint decoding, they proceed with data transmission. The same approach works for the uplink case, as well.

Scheduling transmissions: Another important question is how to select which frames to transmit and who to transmit when the medium is available.

We first answer this question for the downlink traffic. Our goals are to (i) avoid delaying transmissions, (ii) maintain the FIFO order as much as possible, and (iii) maximize the total throughput / parallelism. Therefore we determine the set of frames to transmit by adding the first frame (from the global queue at the controller) to the transmission list. Then we keep adding another frame to the list in the FIFO order as long as (i) the total number of transmissions destinated to a client does not exceed its number of antennas, (ii) it has similar transmission time, and (iii) it does not exceed the maximum number of concurrent frames that can be supported (*i.e.,* $C = min(\sum_i N(i), \sum_j M(j))$, where $N(i)$ is the number of antennas at AP i and $M(j)$ is the number of antennas at client j).

The transmission time of concurrent frames need not be the identical. When a shorter transmission finishes, we consider its intended signal as a null signal for precoding. Moreover, we do not require all transmissions to use the same data rate, since precoding works for analog signals generated from different data rates. In addition, we can possibly concatenate multiple short transmissions so that their total transmission time is similar to that of a long concurrent transmission.

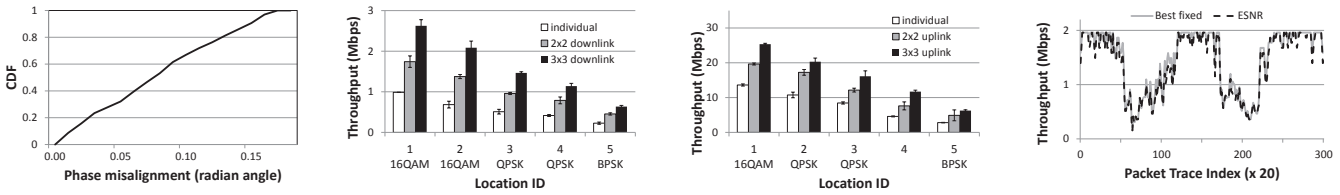
For the uplink, unlike APs, a client has more limited information about traffic at other clients. As a result, upon winning the medium, client j simply randomly selects up to $C - M(j)$ client antennas where C is the maximum number of concurrent frames allowed (*e.g.*, if every client has 1 antenna, it selects $C - 1$ clients; if every client has two antennas, it selects $\frac{C-2}{2}$ clients) to join its transmission. If some selected clients do not have traffic to send at the time, they simply skip transmissions and the APs will detect which clients actually transmit and properly decode the remaining signals by solving a linear system with fewer unknowns.

Obtaining Channel State Information (CSI): For the uplink, the APs get the up-to-date channel estimation from the preambles in the received data frames and we do not need to separately feed back CSI. In comparison, for the downlink, the clients should feed back the CSI to the APs so that the APs can use them for precoding. Our evaluation lets a client feedback CSI once for every five data frames. To further reduce overhead, APs can leverage the channel reciprocity to learn the CSI using the frames received from the clients.

VI. Evaluation

We implement upload multiplexing on SORA for its high data rate. However, SORA cannot support precoding in downlink, because its initial phase offset is random and unpredictable, which makes it infeasible to implementing phase synchronization in downlink. So we implement downlink multiplexing on USRP, which has deterministic initial phase and supports timestamp that allows multiple transmitters to be synchronized in sample level.

Our implementation is based on 802.11a PHY. The uplink communication uses 20MHz bandwidth while the downlink uses 1MHz width. We fix the channel coding to 1/2 rate convolution coding, and use three different modulations (BPSK, QPSK and 16-QAM) that give 6, 12 and 24Mbps transmission rates, respectively in 20MHz bandwidth channel. Frame size is fixed to 1000 bytes for all evaluations.



(a) *Accuracy of phase synchronization* (b) *Downlink throughput* (c) *Uplink throughput* (d) *Rate adaptation in downlink*

Fig. 2. Micro benchmarks and throughput evaluation.

Micro benchmarks: Figure 2(a) shows cumulative distribution function (CDF) of phase misalignment between two USRP transmitters. We measure the amount of the phase error by two senders sending sinusoids separately after the phase synchronization process. The median and 90-th percentile phase misalignments are 0.075 radian and 0.147 radian, respectively. Based on our simulation, these phase misalignments result in 0.4 and 1.2 dB SNR reduction, respectively. This is acceptable given significant multiplexing gain.

Downlink multiplexing: We conduct downlink experiments in 5 placements of APs and clients. At each location, we conduct 5 runs, where during each run every AP sends 1000 frames. To amortize the synchronization overhead, the transmitters send five data packets consecutively after the synchronization process. Figure 2(b) shows the downlink throughput. We evaluate throughput under three rates, but only the result from the rate that gives the best performance is reported for each location under each scheme. Error bars in the figure represent 95% confidence interval. 2×2 and 3×3 downlink represent letting two or three APs send together to two or three clients, while $individual$ represents the case where each AP transmits separately to each client. The throughput increases almost proportional to the number of antennas: the throughput of 2×2 and 3×3 multiplexing are 1.91x and 2.82x of separate transmissions, respectively. It is not exactly 2x or 3x due to slightly increased error rate and overhead.

Uplink multiplexing: Figure 2(c) shows the uplink multiplexing throughput. Here, multiple clients transmit together using the synchronization protocol while APs cooperatively decode the signals. Again we conduct 5 experiments using different placement of APs and clients to evaluate the throughput in various channel quality, and plot the throughput under the best rate for each scheme. The baseline algorithm and Multi-point to Multi-point MIMO have the same best rates in all locations, except the location 4. The average throughput of 2×2 and 3×3 multiplexing are 1.48x and 2.09x of separate transmissions, respectively. It is lower than the number of antennas. This is not an inherent limit of the uplink multiplexing method and can be improved as follows. First, we can compensate for the phase shift caused by CFO as in the downlink to enhance the decoding rate. Second, a better decoding method (*e.g.*, joint decoding) can be used. Third, while we synchronize senders within CP, their OFDM symbols are still slightly misaligned and get increased noise due to Inter Carrier Interference (ICI) from symbol misalignment. The symbol misalignment can be reduced if the scheme is implemented on chipset-level hardware rather than on a software radio that has longer and variable turn-around time. SourceSync [7] showed that multiple transmitters can synchronize within 20 ns using an FPGA implementation.

Rate adaptation: We further evaluate the performance of our rate adaptation scheme described in Section V. We collected QPSK packet traces from 2×2 downlink multiplexing environments. By measuring the dispersion of the symbols in the previous frame, it calculates the BER and determines frame error rate if the signals were modulated as BPSK, QPSK or 16QAM considering FEC; and then select the rate that gives the highest throughput. We compares its performance with the best fixed rate, which uses the traces collected using all modulation for a given location and selects the rate that gives the best throughput. The results in Figure 2(d) show that our rate adaptation achieves 96% throughput of the best fixed rate.

VII. Related Work

There are a few nice experimental work that are closely related to ours. For example, [1] allows a single AP with multiple antennas to send to multiple clients and [11] allows a single AP with multiple antennas to receive data from different clients. [8] and [2] focus on achieving distributed MIMO at the physical layer. Different from [1] and [11], we take a significant step further by allowing different APs to communicate with different clients at the same time, thereby further increasing multiplexing gain. Different from [8] and [2], we go beyond the PHY-layer design by exploring a range of important MAC layer design issues in distributed MIMO.

References

[1] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *Proc. of MobiCom*, Sept. 2010.
[2] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire. Achieving high data rates in distributed MIMO systems. In *Proc. of MobiCom*, 2012.
[3] G. J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas. *Bell Labs Technical Journal*, 1996.
[4] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *Proc. of ACM SIGCOMM*, Aug. 2010.
[5] J. Heiskala and J. Terry. *OFDM Wireless LANs: A theoretical and practical guide*. SAMS, 2001.
[6] P. Murphy. Design, implementation and characterization of a cooperative communications system. *PhD Thesis at Rice University Department of Electrical and Computer Engineering*, 2010.
[7] H. Rahul, H. Hassanieh, and D. Katabi. SourceSinc: a distributed wireless architecture for exploiting sender diversity. In *Proc. of ACM SIGCOMM*, 2010.
[8] H. Rahul, S. Kumar, and D. Katabi. MegaMIMO: scaling wireless capacity with user demands. In *Proc. of ACM SIGCOMM*, 2012.
[9] S. Sen, N. Santhapuri, R. R. Choudhury, and S. ari Nelakuditi. AccuRate: constellation based rate estimation in wireless networ ks. In *Proc. of NSDI*, 2010.
[10] SORA. http://research.microsoft.com/en-us/projects/sora/.
[11] 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 *Proc. of ACM MobiCom*, Sept. 2009.
[12] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[13] USRP. http://www.ettus.com.