Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2009-007 February 18, 2009

Overcoming the Antennas-Per-Node Throughput Limit in MIMO LANs

Shyamnath Gollakota, Samuel David Perli, and Dina Katabi

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Overcoming the Antennas-Per-Node Throughput Limit in MIMO LANs

Shyamnath Gollakota Samuel David Perli Dina Katabi
gshyam@mit.edu samuelp@mit.edu dk@mit.edu

MIT CSAIL

ABSTRACT

Today, the number of concurrent packets in a MIMO LAN is limited by the number of antennas on the AP. This paper shows how to overcome this limit. It presents a new design where multiple client-AP pairs can communicate concurrently, on the same 802.11 channel. We demonstrate both analytically and experimentally that our design almost doubles the throughput of a MIMO LAN.

The key idea underlying our approach is Interference Alignment and Cancellation (IAC), a novel technique for decoding concurrent sender-receiver pairs in MIMO LANs. It exploits two basic properties of MIMO LANs. First, MIMO transmitters can control the alignment of their signals at a receiver. Second, APs are typically connected to a backend Ethernet, which they can use for coordination. Hence, in IAC, transmitters align their signals such that the first AP can decode at least one of the concurrent packets. Once a packet is decoded, it is sent over the Ethernet to the second AP, which subtracts it from its received signal to decode a second packet, which it sends to the third AP to decode the next packet, and so on. We implement our technique in 2x2 MIMO GNU Radios, and demonstrate via wireless experiments that IAC increases the average throughput of a MIMO LAN by 1.5x on the downlink and 2x on the uplink.

1 INTRODUCTION

Multi-input multi-output (MIMO) technology is emerging as the natural choice for future wireless LANs. The current design, however, merely replaces a single-antenna channel between a sender-receiver pair with a MIMO channel. The throughput of such a design is always limited by the number of antennas per access point [3, 33]. Intuitively, if each node has two antennas, the client can simultaneously transmit two packets to the AP. The AP receives a linear combination of the two transmitted packets, on each antenna, as shown in Fig. 1. Hence, the AP obtains two linear equations for two unknown packets, allowing it to decode. Transmitting more concurrent packets than the number of antennas on the AP simply increases interference and prevents decoding. Thus, today the throughput of all practical MIMO LANs is limited by the number of antennas per AP.

In theory, however, one may run many APs on an 802.11 channel, allow multiple clients to transmit concurrently, and have the APs decode the received signals jointly as if they were one giant AP [33, 19], and hence overcome the above throughput limit. This scheme however requires the APs to send the raw received signals over the wired infrastructure (e.g., Ethernet) to a central entity, which jointly solves the linear equations. Since an 802.11 channel spans 20 MHz, each antenna receives 40 Mega samples per second, where each sample is 8 to 16 bits, generating a total of almost half a Gb/s.[1] Thus, to jointly decode three APs with four antennas each, one needs to send 6 Gb/s of traffic on the Ethernet. Such overhead is impractical.

This paper introduces Interference Alignment and Cancellation (IAC), a practical scheme to overcome the antennas-per-node throughput limit in MIMO LANs. IAC exploits that MIMO transmitters can control the alignment of their signals at the receivers [33, 4]. IAC clients align their signals in a manner that allows the first AP to decode one of the concurrent packets. Once a packet is decoded it is sent to the second AP over the wired infrastructure. The second AP, then, uses interference cancellation to subtract the decoded packet from its received signal, allowing it to decode a second packet, which it sends to the third AP to decode the next packet, and so on. Since IAC APs exchange decoded packets, not raw signals, the overhead stays manageable. Hence, IAC presents a practical scheme to support more concurrent packets than there are antennas on any one AP, and overcomes the antennas-per-node throughput limit. In particular, we demonstrate both analytically and experimentally that our design almost doubles the throughput of a MIMO LAN.

To get a better feel for how IAC works, consider again a client that is uploading two concurrent packets. Say we add a second client-AP pair. Can the newly added client-AP pair concurrently upload a third packet? If one allows the three concurrent packets to combine on the medium, without any coordination at the transmitters, each of the two APs gets two linear equations with three unknown packets, and hence cannot decode. IAC, however, exploits that MIMO transmitters can control the alignment of their signals at the receivers. In IAC, the clients encode the packets in a special way to align two out of the three packets at the first AP but not at the sec-

[1] According to the Nyquist criterion, to capture a signal without loss of information one needs to sample it at twice its bandwidth. 802.11 has a bandwidth of 20 MHz. Thus each antenna generates 40 Mega samples per second.

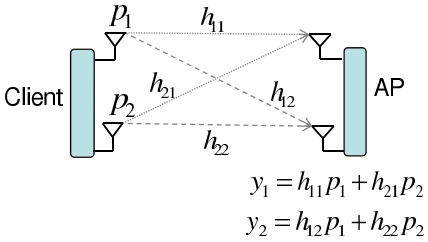


Figure 1—Throughput of current MIMO LANs is limited by the number of antennas per node. The h_{ij}'s refer to known channel coefficients, and the p_i's refer to concurrent packets. The client transmits two concurrent packets on its two antennas. The AP receives a different linear combination of the transmitted packets on each antenna, which it solves to obtain the packets.

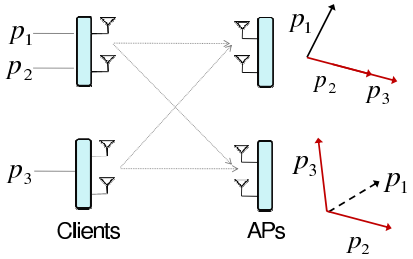


Figure 2—IAC Example. AP1 decodes packet p_1 and sends the decoded packet on the Ethernet to AP2 which then performs interference cancellation to subtract p_1. As a result AP2 can decode p_2 and p_3.

ond, as shown in Fig. 2. As a result, the first AP can treat the second and third packets as one unknown. Hence, the AP has the equivalent of two equations with two unknowns, allowing it to decode the first packet, p_1. The AP then sends the decoded packet on the Ethernet to the second AP, which can now perform interference cancellation to subtract the effect of the known packet. As a result, it is left with two linear equations over two unknown packets, p_2 and p_3, which it can decode. The system delivers three packets per time unit. Hence, its throughput gain is not bounded by the number of antennas per node as it is the case for current MIMO LANs.

Note the synergy between interference alignment and interference cancellation. Interference alignment aligns the interfering packets at the first AP, allowing it to locally decode the first packet and hence bootstrap the decoding process. Interference cancellation enables other APs to use the decoded packet to cancel interference, and hence decode more packets. Neither interference alignment nor interference cancellation alone would be sufficient on its own to decode the three packets in Fig. 2.

IAC has the following key features.

- IAC generalizes to any number of antennas and clients, and applies to both the downlink and the uplink.
- IAC delegates all coordination to the APs, which tell the clients how to encode their packets to produce the desirable alignment. Further, the channel estimates required for computing this alignment can be computed from ack packets with negligible overhead.
- IAC brings in more gains than apparent in the above example. Specifically, for a MIMO system with M antennas, IAC can deliver $2M$ concurrent packets on the uplink, and $\max(2M-2, \lfloor \frac{3}{2}M \rfloor)$ on the downlink – i.e., it doubles the throughput of the uplink, and almost doubles the throughput of the downlink for a large number of antennas.
- IAC works with various modulations and FEC codes. This is because IAC subtracts interference before passing a signal to the rest of the PHY, which can use a standard 802.11 MIMO modulator/demodulator and FEC codes.

We have built a prototype of IAC in GNU radios and evaluated it using a testbed of 20 USRP nodes, each equipped with 2-antennas. Our results reveal the following findings:

- In comparison with point-to-point MIMO, IAC improves the average throughput of a 2-antenna MIMO LAN by 1.5x on the downlink and 2x on the uplink.
- IAC is also as fair as point-to-point MIMO in the sense that no client is worse off than in current MIMO.
- IAC provides a gain for any number of clients including a single active client. In this case, IAC exploits diversity to improve the throughput by 1.2x.

2 Related Work

Related work falls in the following areas.

(a) MIMO Communication Theory. Our work builds on the theory of interference alignment. Recent work has shown that pre-processing signals at the senders in a manner that aligns the interference at the receivers increases the total capacity of wireless networks [5, 8, 13, 20]. This paper however is the first to combine interference alignment with interference cancellation, showing that such a design can improve the throughput of MIMO LANs. This has allowed us to provide gains that traditional interference alignment can not provide on its own (eg., the three packets in Fig. 2). Further, to the best of our knowledge, this paper presents the first implementation of interference alignment, experimentally demonstrating the plausibility of this technique and the benefits of combining it with interference cancellation.

Our work also builds on recent advances in the theory of multiuser MIMO (MU-MIMO). MU-MIMO advocates having multiple clients concurrently communicate with a *single* AP or base station [9, 11, 33, 34]. Thus, the throughput of MU-MIMO is limited by the number of antennas on a single AP [11]. In contrast, this paper overcomes the antennas-per-AP throughput limit. It does so via interference alignment and cancellation, a new approach for decoding concurrent packets in a MIMO LAN.

Our work is also related to prior work on ad-hoc MIMO networks [29, 30, 31, 32]. This prior work presents routing and MAC protocols for MIMO ad hoc networks. Our work however differs from this work both in focus and approach. We focus on MIMO LANs, and we introduce interference alignment and cancellation as a new mechanism to significantly increase the throughput of these networks.

(b) Wireless Networks. This paper is inspired by prior work on cooperative AP design. Past works on single an-

tenna systems has proposed using multiple APs to improve coverage [26, 7], balance the load [21, 22], or recover corrupted packets [24, 35]. This paper builds on this foundation but differs both in focus and mechanisms. This paper focuses on MIMO LANs. It introduces IAC, a new technique that enables MIMO LANs to support a larger number of concurrent transmissions than possible with current designs, hence allowing MIMO LANs to overcome the antennas-per-AP throughput limit.

Prior work has advocated allowing concurrent transmissions in the context of single-antenna nodes. These systems typically *prevent interference* by dividing the resources between users. For example, they might assign the different users different frequency bands [16, 25, 26], or different codes [6, 15]. In contrast, IAC does not assign users different frequency bands or different codes. It uses interference alignment and cancellation to allow users to communicate concurrently in the presence of interference.

Recently, a few projects have advocated the use of interference cancellation in 802.11 networks [12, 17]. Our paper is inspired by this work but it uses interference cancellation in a different context. Specifically, it combines interference cancellation with interference alignment and shows that this new technique brings in a significant throughput gain for MIMO LANs.

Finally, APs with directional antennas divide the space into sectors, each served by a different antenna. This prevents interference between nodes in different sectors, allowing multiple clients to communicate concurrently with the AP. Our approach is orthogonal to directional antennas since we can enable nodes in the same sector (i.e., nodes that interfere) to communicate at the same time.[2]

3 Interference Alignment and Cancellation

IAC's design targets MIMO wireless LANs in a university or corporate campus where APs are connected via a wired infrastructure (e.g., Ethernet). Today these networks use one AP to serve any particular area, and limit interference by assigning adjacent APs to different 802.11 channels. IAC maintains this high-level architecture, except that it replaces every AP by multiple colocated APs that use the same 802.11 channel. Similar to the current architecture, in IAC, adjacent areas employ different 802.11 channels, but different from the current architecture, each of these areas is served by a set of co-located APs rather than a single AP.

IAC allows these colocated APs to serve multiple clients at the same time. It leverages the wired bandwidth to enable the APs to collaborate on resolving interfering transmissions. We show that our approach can transmit almost twice the number of concurrent packets on each 802.11 channel, and hence about double the overall throughput of a MIMO LAN.

[2]It is a common mistake to think that MIMO beam-forming is equivalent to directional antennas. Beam-forming allows the signal to constructively combine at the intended receiver, increasing its throughput. This however still creates interference at nodes that are not in the direction of the intended receiver. Hence, beam-forming cannot overcome the antennas-per-node throughput limit of MIMO LANs.

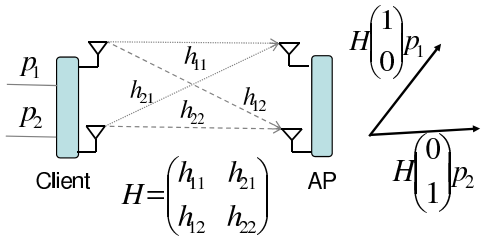


Figure 3—Two Packets on Uplink. The client transmits two packets, p_1 and p_2, from its two antennas. The packets arrive along the vectors $H[1\ 0]^T$ and $H[0\ 1]^T$, where H is the channel matrix and $[.]^T$ refers to the transpose of a vector. To decode p_1 and p_2, the AP projects along the vectors orthogonal to $H[0\ 1]^T$ and $H[1\ 0]^T$ respectively.

IAC has three key components: 1) a physical layer that decodes concurrent packets across APs, 2) a MAC protocol that coordinates the senders and allows the optimal number of concurrent transmissions on the wireless medium, and 3) An efficient mechanism to estimate channel parameters.

4 IAC's Physical Layer

IAC modifies the physical layer to allow multiple client-AP pairs to communicate concurrently on an 802.11 channel. Specifically, IAC operates below the modulation and forward error correcting codes (FEC), and is transparent to both. It is an encoding-decoding scheme that, when applied at concurrent transmitter-receiver pairs, makes each AP-client pair experience a channel that is oblivious to the existence of concurrent transmissions.

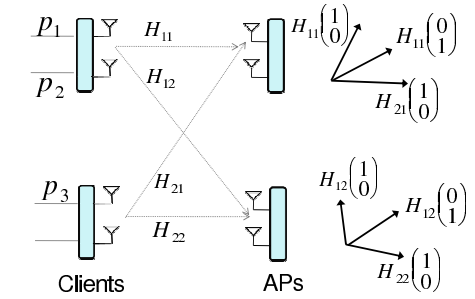
For clarity, we present our ideas in the context of a 2-antenna per-node system, and assume nodes know the channel estimates. Later, we extend these ideas to any number of antennas and explain how we measure channel functions. Our presentation focuses on scenarios where interference from concurrent transmissions is much stronger than noise and is the main factor affecting reception.

(a) Two concurrent packets on the uplink: Let us start with the standard MIMO example in Fig. 3, where a single client transmits two concurrent packets to an AP. Say that the client transmits p_1 on the first antenna, and p_2 on the second antenna. The channel linearly combines the two packets (i.e., it linearly combines every two digital samples of the packets). Hence, the 2-antenna AP receives the following signals:

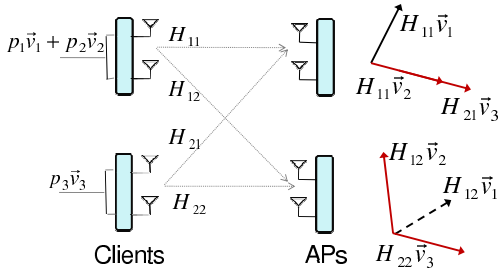
$$y_1 = h_{11}p_1 + h_{21}p_2$$
$$y_2 = h_{12}p_1 + h_{22}p_2,$$

where h_{ij} is a complex number whose magnitude and angle refer to the attenuation and the delay along the path from the i^{th} antenna on the client to the j^{th} antenna on the AP, as shown in Fig. 3.

Since the nodes have two antennas, the transmitted and received signals live in a 2-dimensional space. Thus, it is convenient to use 2-dimensional vectors to represent the system [33]. This representation will allow us to use simple figures to describe how a MIMO system works. We can re-write



(a) Three Packets Without Interference Alignment.



(b) Three Packets With Interference Alignment.

Figure 4—Three Packets with/without Interference Alignment. In (a), the clients transmit the packets without alignment. The packets combine at the APs along three different vectors and the APs cannot decode any packet. The second case shows how IAC delivers three packets on the uplink. Specifically, two of the packets are aligned at AP1, by picking appropriate encoding vectors, allowing decoding of the first packet. AP2, then, subtracts the decoded packet to decode the remaining two packets.

the above equations as:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = H \begin{pmatrix} 1 \\ 0 \end{pmatrix} p_1 + H \begin{pmatrix} 0 \\ 1 \end{pmatrix} p_2, \quad (1)$$

where H is the 2×2 uplink channel matrix (i.e., the matrix of h_{ij}'s). Thus, the AP receives the sum of two vectors which are along the directions $H[1\ 0]^T$ and $H[0\ 1]^T$ (where $[.]^T$ refers to the transpose of a vector), as shown in Fig. 3. Assume the AP knows the channel matrix, H, (we will see how to estimate it in §6). Decoding is easy; to decode p_1, the AP needs to get rid of the interference from p_2, by projecting on a vector orthogonal to $H[0\ 1]^T$. To decode p_2 it projects on a vector orthogonal to $H[1\ 0]^T$. We refer to the direction that a receiver projects on, to decode, as the **decoding vector**.

(b) Three concurrent packets on the uplink: Consider what happens if another client concurrently transmits a packet, as shown in Fig. 4a. Using the same derivation as above, AP1 receives:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = H_{11} \begin{pmatrix} 1 \\ 0 \end{pmatrix} p_1 + H_{11} \begin{pmatrix} 0 \\ 1 \end{pmatrix} p_2 + H_{21} \begin{pmatrix} 1 \\ 0 \end{pmatrix} p_3,$$

where H_{11} and H_{21} are the channel matrices from the first and second clients to AP1. Said differently, AP1 receives the combination of three packets p_1, p_2, and p_3, along three vectors $H_{11}[1\ 0]^T$, $H_{11}[0\ 1]^T$ and $H_{21}[1\ 0]^T$, as shown in Fig. 4a. Since AP1 has only two antennas, the received signal lives in a 2-dimensional space; hence AP1 cannot decode three packets. Said differently, for any packet p_i, the AP cannot find a projection (a decoding vector) that eliminates the interference caused by the other two packets. The second access point, AP2, is in a similar state, it receives the three packets along three vectors $H_{12}[1\ 0]^T$, $H_{12}[0\ 1]^T$ and $H_{22}[1\ 0]^T$, and cannot decode for the same reason.

However, one advantage of MIMO is that a transmitter can control the vectors along which its signal is received. For example, when a transmitter transmits packet p_1 on the first antenna, this is equivalent to multiplying the samples in the packet by the unit vector $[1\ 0]^T$ before transmission. As a result the received vector at the AP is $H[1\ 0]^T p_1$, where H is the channel matrix from transmitter to receiver. If the transmitter, instead, multiplies the packet p_1 by a different vector, e.g., $\vec{v}$, the AP will receive the vector $H\vec{v}p_1$. Thus, instead of transmitting each packet on a single antenna, we multiply packet p_i by a vector $\vec{v}_i$ (i.e., multiply all digital samples in the packet by the vector) and transmit the two elements of the resulting 2-dimensional vector, one on each antenna. Thus, by changing $\vec{v}_i$, we can control the vector along which the AP receives the packet. We call the vector $\vec{v}_i$ the **encoding vector** of packet i.

Now, we can apply this method to the 2-client and 2-AP system to transmit three concurrent packets. In particular, the transmitters multiply packet i with vector $\vec{v}_i$, as shown in Fig. 4b. We want to pick $\vec{v}_2$ and $\vec{v}_3$ such that the second and third packets (i.e., p_2 and p_3) are aligned at AP1, as in Fig. 4b, that is [3] :

$$H_{11}\vec{v}_2 = H_{21}\vec{v}_3,$$

where H_{11} and H_{21} are the channel matrices from the first and second clients to AP1. This can be easily done by picking random values for $\vec{v}_1$ and $\vec{v}_2 (\neq \vec{v}_1)$ and substituting in the above equation to get $\vec{v}_3$ (i.e., $\vec{v}_3 = H_{21}^{-1} H_{11}\vec{v}_2$).[4]

Now that AP1 receives the second and third packets aligned on the same direction as in Fig 4b, it can decode the first packet, p_1, by projecting on a vector orthogonal to the aligned interference, i.e., a vector orthogonal to $H_{11}\vec{v}_2$ and $H_{21}\vec{v}_3$. Since these two vectors are already aligned, there is a vector that is orthogonal to both of them, and thus the AP can decode. Note that without alignment, AP1 could not decode because $H_{11}\vec{v}_2$ and $H_{21}\vec{v}_3$ would have different directions, and no vector will be orthogonal to both.

Note that aligning two vectors with respect to AP1 does not mean that they are aligned with respect to AP2. This is because the channels from the clients to the two APs are different and independent. However, we do not need to align the signals at AP2. AP1 can decode the first packet and send it to AP2 on the Ethernet. Now AP2 knows the first packet. It also knows the channel functions (see in §6 how we compute channel functions). Hence it can reconstruct the signal

[3] In general, aligning the directions would mean $H_{11}\vec{v}_2 = \alpha H_{21}\vec{v}_3$, where α is a scalar. Also note that the vectors are normalized to satisfy the power constraints. But for clarity, we ignore these details in our description.

[4] Channel matrices are typically invertible because the antennas are chosen to be more than half a wavelength apart. If the matrix is not invertible, then you don't really have a MIMO system because the two antennas translate into just one equation.

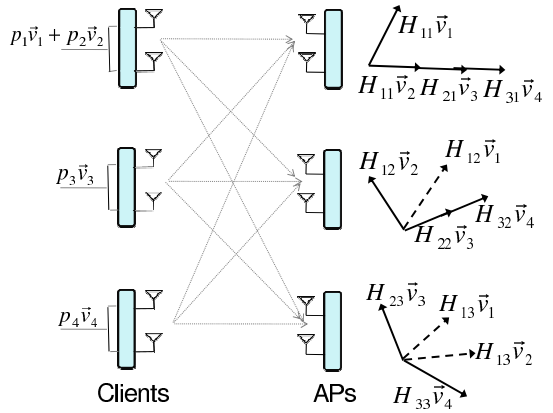


Figure 5—Four Packets on the Uplink. In this figure, IAC decodes one packet each at AP1 and AP2 and two packets at AP3. This requires three packets to be aligned at AP1 and two packets at AP2, which can be done by picking appropriate encoding vectors.

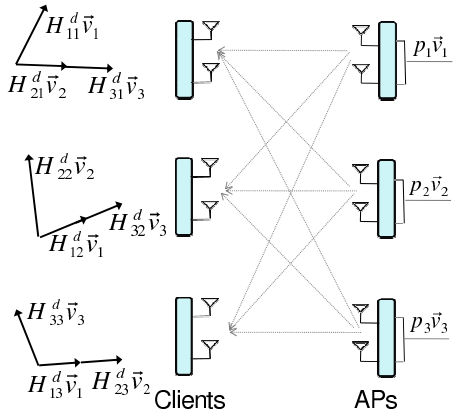


Figure 6—Three Packets on the Downlink. The APs deliver one packet to each client. To enable the client to decode its packet, all the undesired packets at the client must be aligned.

associated with the first packet and subtract it from what it received. This is standard interference cancellation [18, 33]. After cancellation, AP2 is back into a scenario similar to typical MIMO, namely two packets on two different directions, i.e., it is similar to the AP in Fig. 3. Hence, it can decode. Thus, we obtained all three packets. AP1 decoded the first packet, and AP2 decoded the second and third.

(c) Four concurrent packets on the uplink: Let us try to increase the number of concurrent packets on the uplink to 4. We cannot do this with only 2 clients and 2 APs (This is because the system is already too constrained to produce the desirable alignment.) We need to add an additional AP-client pair. For example, consider the three APs and three clients, in Fig. 5. The first client transmits packets p_1 and p_2, the second client transmits p_3 and the third client transmits the fourth packet, p_4. Now that we have developed a vector representation, it is fairly simple to produce an IAC solution for any configuration. Specifically, as shown in Fig. 5, AP1 needs to align 3 out of 4 packets. This results in one free packet, e.g., p_1, which can be decoded with orthogonal projection, as we did earlier. From the perspective of AP2, p_1 is already decoded at AP1, and hence can be subtracted and removed from the signal. Thus, AP2 is left with three unknown packets. To decode one more packet, it needs to have 2 out of 3 packets aligned, as shown in Fig. 5. From the perspective of AP3, two packets are already decoded at AP1 and AP2, and their signal can be canceled using interference cancellation. Thus, AP3 is left with only two unknown packets, which it can decode. Hence, AP3 does not need to align any packets. We can achieve the desired alignment (i.e., the alignment in Fig. 5) by solving the following equations:

$$H_{11}\vec{v}_2 = H_{21}\vec{v}_3 = H_{31}\vec{v}_4 \quad (2)$$

$$H_{22}\vec{v}_3 = H_{32}\vec{v}_4, \quad (3)$$

where H_{ij} is the uplink channel matrix from the i^{th} client to the j^{th} AP. Eqs. 2 ensures the desired alignment at AP1 and Eq. 3 ensures the desired alignment at AP2. Effectively, this translates to three linear equations in three unknowns (the vectors), which can be solved. Thus, the APs can decode four concurrent packets.[5]

(d) The downlink: The discussion so far has focused on the uplink, what about the downlink? Clearly the downlink is more limited, since the clients cannot cooperate over a wired Ethernet. A client cannot decode one packet and send it to other clients for interference cancellation. The lack of cooperation means that the clients have to decode independently. So, we need to align the interference at each client to ensure that it can decode at least one packet. For a 2-antenna system, this means that we can at best deliver 3 concurrent packets on the downlink. This however is still higher than what can be delivered in today's point-to-point MIMO LANs.

Say that we want to deliver packets p_1, p_2, and p_3 to Client 1, Client 2, and Client 3 respectively. Each client needs to receive the two undesired packets aligned along the same vector and the desired packet along a different vector, as shown in Fig. 6. To achieve this behavior, each AP transmits one of the three packets. Now the roles are flipped: the APs are the transmitters and the clients the receivers. Hence, each AP multiplies the transmitted packet by a vector $\vec{v}_i$ that is carefully chosen to ensure the desired alignment. Specifically, we need to ensure:

$$H_{21}^d\vec{v}_2 = H_{31}^d\vec{v}_3 \quad (4)$$

$$H_{12}^d\vec{v}_1 = H_{32}^d\vec{v}_3 \quad (5)$$

$$H_{13}^d\vec{v}_1 = H_{23}^d\vec{v}_2, \quad (6)$$

where H_{ij}^d is the channel from the i^{th} AP to the j^{th} client, i.e., the downlink channels. The three equations above align the packets at each client to ensure that the two undesired packets are along the same vector. These are three linear equations over three unknown vector and can be solved using standard methods (similar to how we solved Eqs. §2 and §3). Hence, each client can decode its desired packet by orthogonal projection.

[5] The solution to the above equation is $\vec{v}_4 = eig(H_{32}^{-1}H_{22}H_{21}^{-1}H_{31})$, where $eig(H)$ is any eigen vector of H, and $\vec{v}_2 = H_{11}^{-1}H_{31}\vec{v}_4$ and $\vec{v}_3 = H_{21}^{-1}H_{31}\vec{v}_4$.

5 MEDIUM ACCESS CONTROL

Since IAC allows multiple clients and APs to transmit simultaneously, it changes the requirements of the medium access control (MAC). The challenge in designing a MAC protocol for IAC arises not only from the need to enable multiple nodes to concurrently access the medium, but also from our desire to maintain minimal complexity at the clients. Specifically, a client should be oblivious to the number of APs in the system, and other clients who might concurrently access the channel. Finally, since traffic is bursty, we need to dynamically change the combination of concurrent clients to match instantaneous traffic demands, while respecting fairness.

The basic principle underlying our solution is to move complexity to the access points, which arbitrate the medium among clients, and also provide each client with its encoding and decoding vectors. Our solution has two components: 1) a MAC protocol that allows multiple nodes to access the medium concurrently, and 2) a concurrency algorithm that decides which clients are served concurrently (on the uplink and downlink).

5.1 Accessing the Medium

Our design extends the 802.11 Point of Coordination Function (PCF) mode to allow it to support multiple concurrent senders. PCF is part of the standard [10]. It allows the AP to arbitrate the medium by polling the clients, and is originally designed to enable 802.11 networks to deal with time sensitive information.

(a) Contention-Free and Contention Periods. In IAC, one of the APs is designated as the leader. The leader AP acts as a coordinator. It polls the clients and grants access to those who have data to transmit [10]. Similar to PCF, we divide time into: Contention Free Period (CFP) and Contention Period (CP), as shown in Fig. 7. A contention-free period starts with the leader AP broadcasting a beacon that announces the duration of the current CFP. During a CFP, the leader AP coordinates access to the medium enabling the nodes to transmit using IAC. This is followed by a contention period, during which any node can contend for the channel using standard 802.11 and point-to-point MIMO.

The objective of this design is to use the contention period to allow new clients to associate with the APs, or to transmit after a long period of silence, using point-to-point MIMO. In contrast, the contention-free period (CFP) is used to pack transmissions as much as possible, increasing throughput. The duration of the contention period (CP) is constant, while the duration of the CFP varies depending on congestion. During a CFP, the APs serve one packet (on the uplink and downlink) to each client that has pending traffic. Hence, when congestion is low and queues are empty, the CFP naturally shrinks, and clients spend more time in CP. When congestion is high, many clients have pending traffic and hence the CFP expands, which is desirable as this mode uses IAC to pack transmissions and increase efficiency.

(b) Acquiring the Medium During CFP. Next, we explain how concurrent transmitters acquire the medium during a CFP. Clearly, this requires knowing which clients are served concurrently. This is the job of the concurrency algorithm, which divides the clients with pending traffic into groups of concurrent transmissions that we call **transmission groups**. It further decides which AP serves which client in a transmission group, and the values of the encoding and decoding vectors. The process for making this decision is described in §5.2. In this section, we focus on how to deliver the packets in each transmission group once the decision is made.

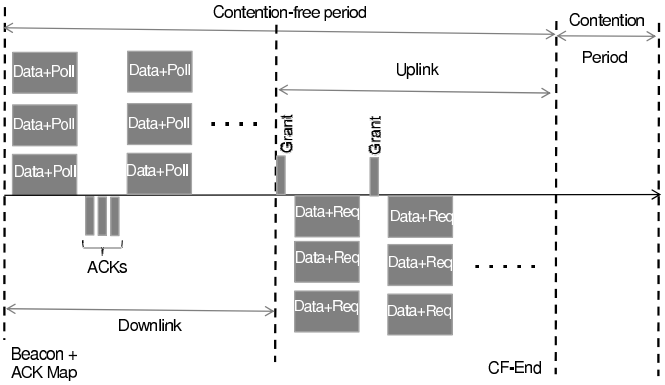


Figure 7—IAC's Extension to PCF.

Figure 8—The metadata in a `DATA+Poll` frame. This metadata is broadcast by the leader AP alone to inform the clients in a downlink transmission group of their decoding vectors and the other APs of their encoding vectors.

Fig. 7 shows the series of events during a contention-free period. At the beginning of a CFP, the leader AP sends a beacon. The leader AP then steps through the downlink transmission groups, one at a time, transmits their downlink packets with the help of other APs, and polls the corresponding clients for uplink traffic. This mode is similar to current PCF behavior, except that in the current PCF the AP steps through a list of individual clients, one at a time; whereas in IAC the leader AP steps through a list of transmission groups.

(b.1) Downlink. The leader AP first goes through the list of downlink transmission groups. With the help of other APs, it sends a `DATA+Poll` frame to each group. This frame has two parts. The first part, shown in Fig. 8, is broadcast by the leader AP alone, and contains the ids of the clients in the group and their encoding and decoding vectors. The ids are given to the clients upon association. The encoding and decoding vectors are computed by the concurrency algorithm which runs on the leader AP. The leader AP also includes a frame id, `Fid`, the number of APs and a checksum of its broadcast. Further, it sets the length of the `DATA+Poll` frame to the maximum length of the packets in the transmission group, so that all clients know when the frame ends. The second part of the frame is the combination of concurrent transmissions by all APs. For the example of three APs with 2-antennas each, this part has the three APs transmitting a packet to each of the three clients in a transmission group.

Note that both the clients and APs listen to the leader

AP as it broadcasts the first part of the DATA+Poll frame. The APs need to learn their encoding vectors since without these vectors they cannot transmit concurrently to the clients. The clients need to learn the decoding vectors to be able to decode their data. The clients and APs can use the checksum to test whether they received the correct information. Note that the transmissions still work fine if any of the APs or the clients failed to hear the leader AP. Specifically, the AP/client who failed to hear the leader AP, will not transmit. The other transmissions can go as desired.

After the DATA+Poll frame, the clients in the transmission group send their acks, one after the other, using traditional MIMO. The order in which they transmit these acks is the same as the order of their ids in the DATA+Poll frame. These acks are similar to synchronous 802.11 acks. In 802.11, they are sent one after each data packet. Here the data packets are sent concurrently and all acks follow.

(b.2) Uplink. After going through all downlink groups, the leader AP steps through the uplink groups. Similar to the downlink case, the leader AP first broadcasts a Grant frame specifying the ids of the clients that will transmit on the uplink, and the encoding and decoding vectors. The other APs silently listen to the encoding and decoding vectors and wait for clients' transmissions. The clients in an uplink group use their encoding vectors to transmit simultaneously on the uplink. Each client transmits a Data+Req frame. This frame contains the client's uplink data. If the client still has traffic to send, the frame will also contain a new request for transmission. Each AP listens to the Data+Req frame and projects the received signal on the proper decoding vector. This projection is orthogonal to the interfering signals and hence it allows each AP to receive its client of interest.

One difference between the uplink and downlink is that, while each client on the downlink can immediately ack the reception of its packet, the APs need to decode successively using interference cancellation and hence cannot send synchronous acks. The solution however is simple. During the following contention period, the APs inform the leader AP of successful receptions using Ethernet. The leader AP combines this information and sends all acks at the beginning of the next CFP, by embedding them in the beacon information as a bit map. This should not cause any important delay since it allows all clients in the CFP mode to learn the status of the previous packet before they get to send the next packet.

At the end of CFP, the leader AP sends a CF-End frame. This allows the clients to go back to the contention mode, where they use traditional point-to-point MIMO.

A few points are worth noting.

(a) How much synchronization do we require between concurrent transmitters? We do not require concurrent transmitters to synchronize their transmitted symbols. The synchronization between transmitters is loose and it is intended for being efficient. We do not however expect their signals to combine in a particular way at the symbol level. This is because once a receiver projects on its decoding vector, it will receive only the signal from the intended transmitter. Interference from the other transmitters is orthogonal to the decoding vector, and hence is irrelevant. This argument does apply to both the uplink and downlink.

(b) How do we deal with lost packets and retransmissions? If a packet is lost on the uplink, the client discovers the loss from the lack of an ack (at the beginning of the next CFP) and asks for a new transmission slot next time it is polled. If a packet is lost on the downlink, the corresponding AP discovers the loss immediately, from the lack of a client ack, and asks the leader AP to schedule a retransmission.

(c) Is it possible for various APs to make inconsistent decisions? Only the leader AP makes decisions, while the other APs are dumb transmitters/receivers. Similar to the clients, they receive their encoding and decoding vectors for each transmission group over the medium and use them without any modification. They only inform the leader AP in case a packet is lost, or the channel's estimate has changed.

(d) How often do the APs need to communicate over the Ethernet and what do they exchange? As described in §4, the APs perform successive interference cancellation by exchanging the decoded packets over the Ethernet. Further, the subordinate APs need to tell the leader AP whenever a packet is lost or the channel coefficients to a client changes by more than a threshold value. The APs can send this information as annotation on the packet they exchange to perform interference cancellation.

(e) How large is the Ethernet overhead? To minimize Ethernet overhead, IAC connects the set of colocated APs using a hub. This design ensures that every decoded packet is broadcast only once to all colocated APs and to the switch that forwards the packet to its wired/final destination. In this design every packet is transmitted once and there is no extra overhead. While a hub is less efficient for a general Ethernet than a switch, it is a natural choice to connect each set of colocated APs. This hubbed network is then connected to the rest of the Ethernet via a switch.

(f) How large is the wireless overhead associated with IAC's MAC? IAC introduces metadata to coordinate clients and APs. Specifically, concurrent transmissions are preceded by a short broadcast from the leader AP to inform the client-AP pairs of their encoding and decoding vectors. Such a broadcast message already exists in 802.11 PCF mode. [6] We only annotate these messages with extra information. The information that we send is a few bytes per client-AP pair. Assuming 1440 byte packets, the overhead of the metadata amounts to 1-2%. In comparison, the throughput improvement expected from IAC is 1.5x to 2x, which more than compensates for the loss.

[6] Existing 802.11 calls the Grant frame CF-Poll, i.e., it is a poll without downlink data.

5.2 Concurrency Algorithm

The concurrency algorithm runs at the leader AP. The leader AP maintains a FIFO queue for traffic pending for the downlink and a similar queue for uplink requests learned from `DATA+Poll` frames (see §5.1). Given the queues of uplink and downlink traffic, the concurrency algorithm generates the uplink and downlink transmission groups. Without loss of generality, we will focus on the downlink.

There are multiple options for how to combine clients. The brute force approach considers all combinations of clients with queued packets and all different ways of assigning them to the existing APs, computes the encoding and decoding vectors, and estimates the throughput of each combination.[7] It then creates transmission groups for the queued packets in a manner that maximizes throughput. There are two problems with such approach. First, estimating the throughput for every combination of clients in the queue is a combinatorial problem in the number of clients. Second, since this approach focuses on maximizing the throughput, it always prefers to transmit to clients with good channels and hence results in deep unfairness. Alternatively, one can always create transmission groups by combining packets according to their arrivals in the FIFO queue. This approach is simple and gives each client a fair access to the medium, but is completely oblivious to the throughput of a particular grouping. In practice, different groups may yield significantly different throughput gains (see §8.3).

(a) The Best of Two Choices. IAC's concurrency algorithm balances the desire for high throughput with the need to pick a fair group assignment. To prevent starvation and reduce delay, it always picks the head of the FIFO queue as the first packet in the current transmission group. To reduce computational overhead as it picks the rest of the clients in the group, it uses a randomized design that exploits the best of two choices, a standard approach for reducing the complexity of combinatorial problems [23]. Say each group has three clients, and we already have picked the first client in the group as the client whose packet is at the head of the transmission queue. We randomly pick two clients with queued packets as candidates for the second position in the group. Similarly, we also randomly pick two clients for the third position in the group. Now we estimate the throughput for the four transmission groups formed by these potential candidate clients and pick the group that optimizes the throughput. As a result, instead of computing the throughput for every possible combination of clients, we just compute it for four random client combinations.

Next, let us consider the fairness of the approach. A client is considered for transmission either because it is at the head of the queue or because of a random choice. Both of these cases give the client a fair access to the medium. However, since after picking the candidate clients, we still optimize for the overall throughput, we need a mechanism to ensure that clients that never maximize the throughput of a group get picked. To do this, we assign a credit counter to each client. If the client is considered as a result of a random choice, and is ignored because it does not maximize the grouping throughput, the counter is incremented; but if the client is picked for transmission the counter is reset. If the counter crosses a threshold, the client is selected as part of the transmission group irrespective of the overall throughput. This mechanism ensures that every client is part of some group at least a minimum number of times.

[7] The throughput of a transmission group can be estimated as follows: $\sum_i \log(1 + ||\vec{v_i}^{\dagger} H_i \vec{w_i}||^2)$, where the sum is over client-AP pairs, H_i is the channel for a pair, and $\vec{v_i}$ and $\vec{w_i}$ are the corresponding encoding and decoding vectors [33]. The vectors are computed by solving the equations in §4 and the channel matrix is estimated as described in §6.

6 Channel Estimation

In IAC, the APs collaboratively estimate the channels and convey this information to the leader AP as annotation on the decoded packets sent over the Ethernet.

(a) Uplink: To estimate the encoding and decoding vectors for the uplink, we need the physical channel from each concurrent client to each AP, as shown in Eqs. 2 and 3. In the absence of concurrent clients, estimating this channel is a standard MIMO technique [2]. Thus, the first time a client broadcasts an association message, all APs estimate the channel from that client to themselves. (in IAC APs are co-located and hence should all hear the association message or its retransmission. The association does not succeed until they all hear it.) Once the APs have an initial estimate, they need to track it. This is done using the client's ack packets from the contention-free period, and its data packets from the contention period. Both packet types are transmitted without any concurrent transmissions. Hence they can be processed using standard MIMO channel estimation algorithms [2].

Since the APs can estimate the channel from every ack the client transmits, they obtain a frequent estimate of the channel. In static environments the channel is relatively stable and can be easily tracked at this estimation frequency. Slight inaccuracy in estimating the channel only means that the interference is not fully eliminated after applying the encoding and decoding vectors. As long as most interference is eliminated, the loss in throughput stays negligible.

(b) Downlink: Channel estimation is typically done at the receiver [6]. Thus, we have two options: either have the clients estimate the channels and convey this estimate to the leader AP when it polls them, or try to have the APs estimate the channel by exploiting reciprocity between uplink and downlink channels. In our measurements, the latter option worked with sufficient accuracy and hence we adopt it. Reciprocity means that the channel from node A to node B is the transpose of the channel from B to A. Thus, an AP can use the uplink channel from a particular client to infer the downlink channel to that client.

It is important to understand that channel reciprocity does not mean that the link between two nodes A and B is symmetric. Reciprocity (i.e., the kind that we care about in this paper) means that the channel coefficients are the same, but the

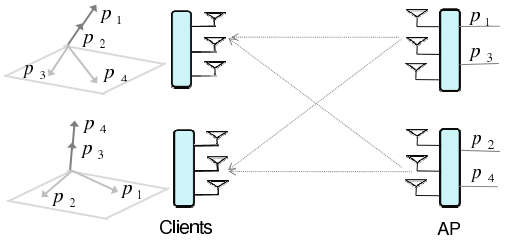


Figure 9—Four Packets on Downlink. At the first client, packets p_3 and p_4 are aligned along one dimension, allowing p_1 and p_2 to lie in a two dimensional space and hence be decoded. Similarly, at the second client, packets p_1 and p_2 are aligned, allowing p_3 and p_4 to be decoded.

noise or the interference could be vastly different. For example, if A transmits symbol x, node B receives $y_B = Hx + n_B$. If we transmit in the opposite direction, node A receives $y_A = Hx + n_A$. The channel multiplier, H, is the same, but the noise could be much higher at A if it is close to a microwave oven. Hence, one may see many packet drops at A but not at B, but this observation does not contradict reciprocity. Reciprocity has been confirmed in measurements [14, 28, 27] and is used in QUALCOMM's 802.11n proposal [2].

Reciprocity cannot be applied directly without calibration to account for hardware differences between the tx and rx chains. The calibration however can be computed once and does not change for the same sender receiver pair. IAC uses a calibration method from the QUALCOMM's 802.11n proposal [2]. Let H^d be the channel between a particular AP and client pair, and H^u the uplink channel from that client to the same AP. Then:

$$(H^d)^T = C_{Client,rx}\, H^u\, C_{AP,tx}, \tag{7}$$

where H^T refers to the transpose of H, and $C_{Client,rx}$ and $C_{AP,tx}$ are constant diagonal matrices that describe the extra attenuation and delay observed by the signal in the transmit and receive hardware chains on the client and the AP respectively.

7 Beyond Two Antennas

This section explores the performance limits of IAC. Specifically, for the general case of M antennas per-node, what is the maximum number of concurrent packets that can be delivered? Further, how many APs are needed to support such a system?

It is important to grasp the constraints in a system like IAC. Naively, it might seem that the number of concurrent packets is constrained only by the number of APs. Specifically, it might seem that one can align the received packets at every AP, allowing each of them to decode at least one packet, and hence one can keep increasing the number of concurrent packets by increasing the number of APs. This is however misleading because aligning a signal at one receiver limits the ability of the transmitter to freely align it at a second receiver. In particular, every alignment imposes new constraints on the encoding vectors at the transmitter. For a feasible solution, the constraints should stay fewer than the free variables in an encoding vector. Since the encoding vector has as many variables as there are antennas on the node, the number of the constraints cannot exceed the number of antennas. Thus, using more APs is beneficial but only up to some point, after which one needs to increase the number of dimensions of the system, i.e., the number of antennas. Below, we demonstrate that in IAC, the number of concurrent packets can be almost twice the number of antennas, and that this gain is achieved with a relatively small number of APs.

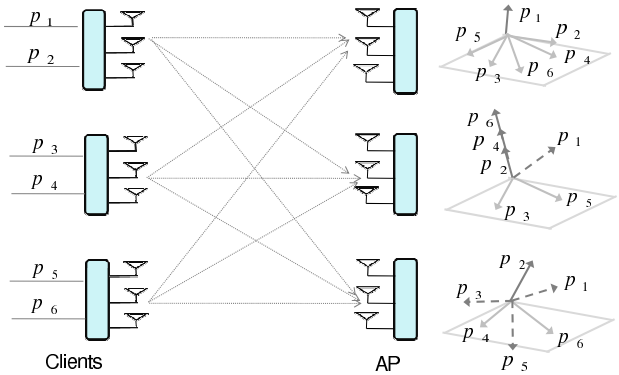


Figure 10—Six Packets on Uplink. At AP1, all the packets other than p_1 are aligned on a two dimensional plane, allowing p_1 to be decoded. At AP2, p_2, p_4 and p_6 are aligned along, allowing p_3 and p_5 to be decoded. At AP3, we cancel p_1, p_3 and p_5 leaving p_2, p_4 and p_6 to lie along three different dimensions and be decoded.

(a) Downlink. In the appendix, we prove the following:

LEMMA 7.1. *In a system with M antennas per node, the maximum number of concurrent packets IAC can deliver on the downlink is* $\max\{2M-2, \lfloor \frac{3}{2}M \rfloor\}$ *and IAC achieves this with* $\max(3, M-1)$ *APs.*

For $M = 3$, the above lemma tells us that we can achieve 4 concurrent packets on the downlink. Fig. 9 shows the downlink case. We have two APs and two clients. Each AP transmits two packets, one for each client. Since the clients have three antennas, the signal is in a three dimensional space. Thus, if we align two packets along one dimension, the other two packets are free of interference and can be decoded, resulting in 4 concurrent packets.

The above procedure can be generalized to any number of antennas. Specifically, if we have $M-1$ APs and two clients, a procedure that makes each AP transmit a packet to each client can deliver a total of $2M-2$ concurrent packets across the two clients. For a large M, this almost doubles the throughput of current MIMO LANs.

(b) Uplink. In the appendix, we prove the following:

LEMMA 7.2. *For a M-antenna system, three or more APs, and at least two clients, IAC can deliver* $2M$ *concurrent packets on the uplink.*

For $M = 3$, the above lemma tells us that we can achieve 6 concurrent packets on the uplink. Fig. 10 shows three clients transmitting to three APs. At the first AP, five out of six packets are aligned in the same plane. This leaves one packet free of interference and hence can be decoded. From the perspective of the second AP, one packet is already decoded and hence can be eliminated from the received signal.

Out of the five packets left, the second AP needs to have three packets aligned along one dimension and two free packets, allowing it to decode two packets. Finally, from the perspective of the last AP, three packets are already decoded and hence their interference can be eliminated. This leaves the last AP with three unknown packets in a three dimensional system and hence it can decode all of them.

Again, this procedure can be applied independent of the number of antennas. Specifically, one needs to align $2M$ packets such that the first AP can decode one packet, the second AP decodes $M-1$ packets and the last AP decodes M packets.

8 Performance Evaluation

We evaluate IAC in a testbed of MIMO software radios. Each node is a laptop connected to a 2-antenna USRP radio board and runs the GNU-Radio software. To create a MIMO node, we have equipped each USRP with two RFX2400 daughterboards. We also set the MUX value in software to allow the FPGA to process samples from both antennas.

(a) Topology. The topology of our network is shown in Fig. 11. It has 20 nodes, three of which are access points. The access points are colocated, showing that our gains are not due to spatial reuse. Moving the access points away from each other makes it easier for IAC since it decreases the interference of concurrent client-AP pairs.

(b) Modulation. IAC uses the modulation/demodulation module as a black-box and works with a variety of modulation schemes. Our implementation, however, uses *BPSK*, which is the modulation scheme that 802.11 uses at low rates.

(c) Configuration Parameters. We use the default GNU Radio parameters. However, in order to drive two antennas at the same time, we double the interpolation and decimation rates at the transmitter and the receiver. Given the above parameters and a BPSK modulation, the resulting bit rate is 250kb/s/antenna. Each packet consists of a 32-bit preamble, and 1500-byte payload.

(d) Compared Schemes. We compare the following:

- **IAC:** This is our implementation of interference alignment and cancellation.
- **MIMO:** There are multiple proposals for 802.11n [2, 1]. These schemes are all point-to-point, i.e., they allow only one transmitter to access the medium at any point in time. They however differ in the amount of channel information available to the transmitter, with more channel information leading to better performance [33]. Since IAC uses full channel information, we compare it with a MIMO design with full channel information available to both sender and receiver. This design is based on QUALCOMM's eigenmode enforcing [2] and uses an approach that is proven optimal for point-to-point MIMO [33].

Three points are worth noting.

- First, we allow the MIMO baseline access to the same number of APs as IAC. MIMO cannot use the additional APs for concurrent transmissions, however it can use them to increase diversity. For example, if there are three APs, each MIMO client can communicate with the AP to which it has the best SNR.
- Second, we use a simplified TDMA MAC for both IAC and the baseline MIMO. The MAC assigns the same number of transmission timeslots to MIMO and IAC. Consider an uplink scenario that involves three clients and three APs. We start with the MIMO experiment and assign each client to its best AP. We successively assign 100 time slots to each client, for a total of 300 time slots for the MIMO experiment. We follow with an IAC experiment where clients transmit together for a total of 300 time slots. We then repeat the experiment for a different client set. This simplified MAC allows for a fair comparison between IAC and the baseline MIMO because it assigns the medium equally to each scheme. Implementing the MAC in §5 requires access to accurate timing information, and the ability to quickly switch the board from a transmit mode to a receive mode. These requirements are not supported by the current USRP-GNURadio platform.
- Finally, both IAC and the baseline MIMO use the GNU-Radio basic decoding modules (e.g., packet detection, clock recovery, synchronization, and channel estimation) and the same system parameters.

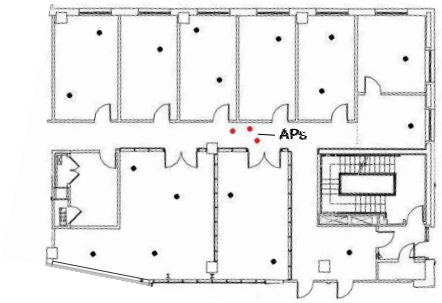


Figure 11—Testbed Topology.

(e) Metric. It is typical in the networking community to compare the throughput of various designs. Throughput results, however, do not bring much insight for radios that do not have proper rate adaptation. Specifically, both in theory and practice, wireless systems (e.g., 802.11a/b/g/n cards, WiMax, etc.) can exploit a higher SNR to use denser modulation and coding schemes and hence increase their throughput. Current GNURadios however do not yet support rate adaptation. Further, they do not have a stable implementation of the set of modulation and coding schemes that allow rate adaptation. In such systems it is unfair to compare throughput because two systems may have the same throughput yet one of them has a significantly higher SNR. In an actual wireless product, the higher SNR system could have used better modulation and coding schemes to achieve a higher throughput. Another way to look at the problem is as follows. Say we take a 2-antenna MIMO system and we show that we can decode four concurrent packets, while a baseline MIMO decodes only 2 concurrent packets. In this case the through-

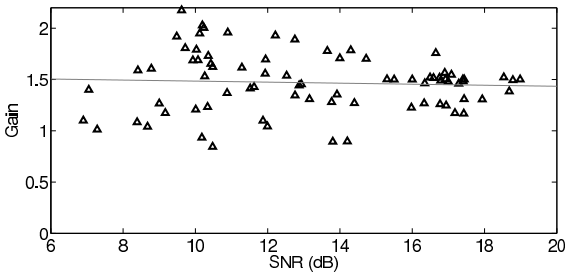


Figure 12—2-clients and 2-APs uplink. In this case, IAC achieves an average gain of 1.5x over traditional MIMO.

put of our system will be double the throughput of the baseline MIMO. This result however is ambiguous since it is not clear whether the MIMO system has a higher SNR. If it does, then the MIMO baseline could have used denser modulation and coding schemes, potentially doubling its throughput, or maybe tripling it. Because of this ambiguity, it is common to measure performance at the physical layer in terms of SNR or a function of it.

Specifically, we can measure the best rate that a particular design can achieve as [33]: $Rate = 10\log_{10}(1+SNR_{Measured})$. In contrast to the measured throughput, the best rate captures the ability of a higher SNR scheme to achieve a higher rate if allowed to scale the modulation and coding schemes with the received SNR. Given the best rate of each design, we can compare IAC against a baseline MIMO system by comparing their best rates as follows:

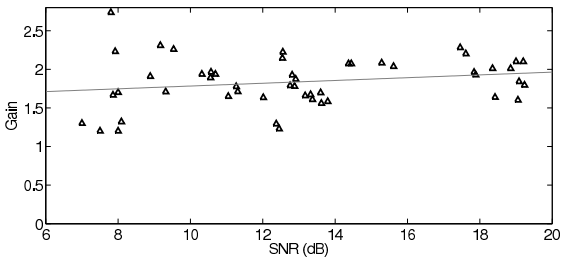
$$Gain = \frac{Rate_{IAC}}{Rate_{MIMO}}. \quad (8)$$

8.1 IAC's Multiplexing Gain

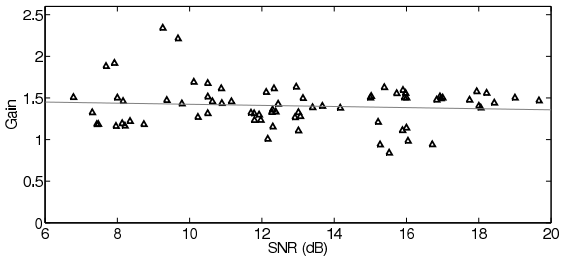
The main advantage of IAC is that it increases the number of concurrent packets, i.e., it provides a multiplexing gain. In §7, we demonstrated this multiplexing gain analytically. Here, we check the gain in practice.

Experiment 2-by-2 Uplink. We randomly pick two clients in the testbed to upload traffic to two APs. We compare IAC to the baseline MIMO where each client uses the best AP. In the baseline MIMO, each client can transmit two packets simultaneously, but the two clients cannot access the medium at the same time and have to alternate in using the medium. In contrast, IAC should be able to deliver three packets simultaneously, as we showed in §1. Specifically, in the first time slot, client 1 uploads a single packet, whereas client 2 uploads two packets. In the next time slot, the two clients alternate, i.e., client 1 gets two packets and client 2 gets one packet. We repeat the experiment many times with different clients. We compute the gain as the ratio between the sum of the rates of the two clients with IAC, to the sum of their rates with baseline MIMO.

Results for 2-by-2 Uplink. Fig. 12 plots the gain of IAC over the baseline MIMO. The x-axis is the average SNR across the two clients in the MIMO experiment, i.e., the average SNR that a client gets to its best AP. The SNRs range from 6 dB to 20 dB which covers most of the operational range of 802.11a/g.[8] Each point refers to an experiment. The dotted line is the linear regression of the points.



(a) Uplink.



(b) Downlink.

Figure 13—3-clients and 3-APs. In this case, over a range of SNRs, IAC provides an average gain of 1.8x on the uplink and 1.4x on the downlink.

The figure supports the analysis, showing that for the 2-client and 2-AP uplink case, IAC increases the throughput by an average of 1.5x. It also shows that there is a significant variance in the gain, particularly at low SNRs. Differences in the gain come from relative differences between the channels of the two clients. To see why this is the case consider the extreme scenario when the two clients have exactly the same channels to the two APs (i.e., $H_{11} = H_{21}$ and $H_{12} = H_{22}$). In this case, aligning the two clients at one AP implies aligning them at the other AP, and hence you cannot decode. In practice, the two clients are extremely unlikely to have the same channel to both APs. However, the closer the channel matrices are the more the gain is affected by noise and imperfection of the channel estimates. This effect is greater at low SNRs where the noise power is higher.

Experiment 3-by-3 Uplink and Downlink. Next, we want to check whether IAC can further increase the multiplexing gain. In §7, we found the bound on the number of concurrent packets. Since our nodes have 2 antennas each, we expect IAC to multiplex 4 packets on the uplink and 3 packets on the downlink. We examine whether our implementation can deliver these gains.

Each experiment involves three clients and three APs, and is run for MIMO and then for IAC. In the MIMO experiments, each client accesses the medium alone and uploads/downloads 2 packets per timeslot. The medium is arbitrated between the three clients. In IAC, all three clients access the medium concurrently. In downlink experiments, each client transmits 1 packet per timeslot. In uplink experiments, in every timeslot, one of the clients transmits 2 packets, the other clients transmit one packet each. We choose

[8] Beyond this range, the USRP receiver starts saturating, i.e., becoming nonlinear.

the client that transmits the two packets in each timeslot in a round robin manner.

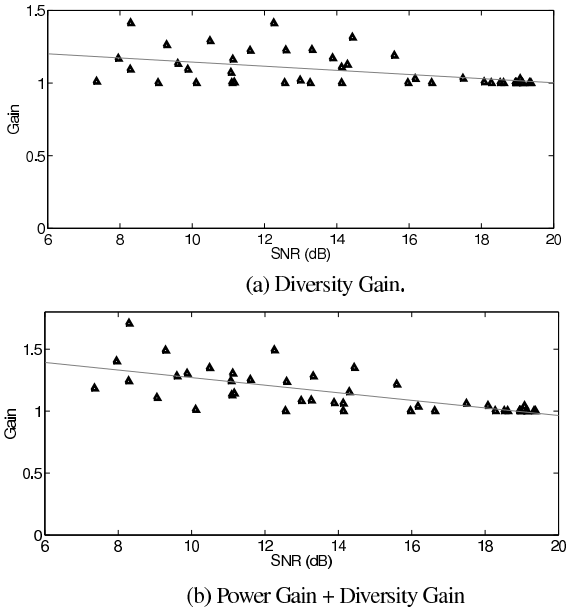
Results for 3-by-3 Uplink and Downlink. Figs. 13a and 13b plot IAC's gain over the baseline MIMO. The x-axis is the average SNR across the three clients in the MIMO experiment, i.e., the average SNR that a client gets to its best AP. The figures show that IAC delivers the expected multiplexing gain. Specifically, it provides about 1.4x increase in transfer rate on the downlink and 1.8x the rate on the uplink. Furthermore, the figure shows that these gains are achieved across the whole SNR range (6 dB or higher).

8.2 IAC's Diversity and Power Gains

Our discussion so far has focused on multiplexing clients to provide a throughput gain. It is natural to wonder whether IAC helps in networks with very few clients. In particular, what if there is only one client? In this case, there is no multiplexing gain, i.e., the maximum number of concurrent packets that can be communicated to/from one client is two (since it has 2 antennas).

However, because of its ability to coordinate multiple APs over the Ethernet, IAC can deliver two other types of gains: a diversity gain and a power gain. The diversity gain arises from the ability to choose between various antennas. For example, consider the downlink when there is one client and two APs. We want to deliver two concurrent packets to the client. The baseline MIMO exploits diversity by selecting the best among the two APs.[9] IAC however has more options because it can exploit diversity across APs; for example it can use one antenna from each AP or use all four antennas together or any option in between.[10] The power gain comes from using multiple concurrent transmitters. Since each node has a limit on maximum power, transmitting two packets from the same node means that the node has to divide the power between them. In contrast, transmitting the two packets from different nodes, means that each packet can be transmitted at full power.

Experiment with 1-client and 2-APs. Every experiment uses one random client and two APs. The client downloads 100 packets. We compare a MIMO design where the client downloads its packets from the best AP (i.e., the AP that delivers the highest SNR to the client) with a IAC design where the two APs cooperate on downloading the packets to the client. Specifically, the leader AP compares the following options: assign one packet to each AP, and assign both packets to one of the two APs. It picks the option that has a better throughput. In both MIMO and IAC, two packets are transmitted simultaneously in every timeslot. For each client choice, we repeat the experiment with IAC under two power settings: (1) Each IAC AP has the same power as in the MIMO run, and (2) Each IAC AP has half the power used by the APs in the MIMO run.



(a) Diversity Gain.

(b) Power Gain + Diversity Gain

Figure 14—1-client and 2-APs. In this case, IAC provides two types of gains. Figure (a) shows the diversity gain that come from using multiple APs to concurrently transmit two packets. Figure (b) shows that exploiting the power gain from using concurrent transmit nodes further increases the gains.

Results with 1-client and 2-APs. Figs. 14a and 14b plot the increase in download rate achieved with IAC in comparison with the baseline MIMO. The figure shows that IAC outperforms MIMO for both power settings. More specifically, the results reveal the following:

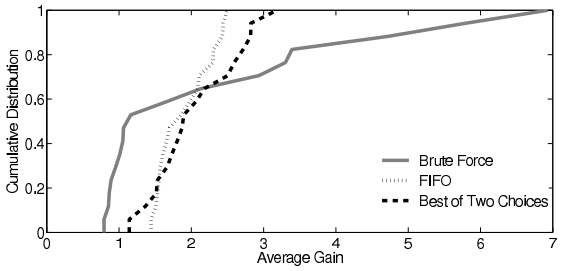
- IAC is beneficial even when the network has only one active client.
- IAC provides a higher diversity gain than current MIMO. Even if the APs are forced to reduce their transmission power such that the sum of the consumed power stays the same for a MIMO LAN and IAC LAN, IAC can deliver a higher rate to the client. This is because the baseline MIMO can choose only between APs, but IAC can exploit antenna diversity across APs.
- IAC can also increase the rate by exploiting the power gain from using concurrent transmit nodes. Comparing Figs. 14a and 14b, we see that the additional gain from increased power is around 10%-18% at low SNRs.
- Most of the gains come at the lower end of the SNR spectrum. This is expected since doubling the power or having two diverse choices gives an SNR improvement of about 1-3 dB [33]. This translates to high gains at low SNRs, but relatively low gains at higher SNRs. Thus, the diversity and power gains play a role at low SNRs but the multiplexing gains are dominant at high SNRs.

8.3 IAC in a Large Network

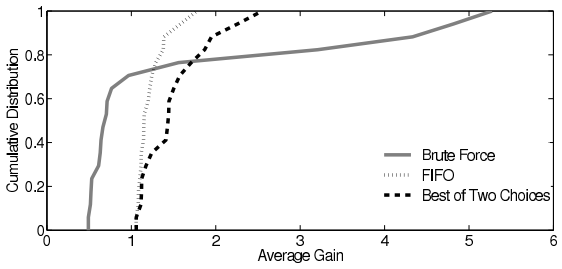
We investigate IAC's performance in a large network with many active clients (e.g., a large conference room).

[9] The literature presents a second type of diversity gain for point-to-point MIMO. In this case, the two antennas are used to send/receive the same packet/symbol. This however is a weaker gain since it requires MIMO to give up its multiplexing gain and repeat the same data on both antennas [33].

[10] Note that comparing the options to find the best can be done merely by computing the capacity using our knowledge of the channel matrices [33].



(a) Uplink.



(b) Downlink.

Figure 15—Gains achieved for the Whole Testbed. The figure shows CDFs of client gains for three concurrency algorithms, both for the uplink and the downlink. The CDF is taken over 17 clients all of which have traffic to send/receive. All concurrency algorithms provide significant throughput gains on average, but the *best-of-two* algorithm has the best fairness-throughput tradeoff.

When the number of clients is larger than the maximum number of concurrent packets, one has many options for which clients transmit concurrently, both on the uplink and downlink. Choosing a particular option impacts both fairness and the total throughput. In fact, in any wireless network, there is always a tension between maximizing throughput and ensuring fairness because the best option in terms of throughput would always transmit to the client with the best channel and starve the others. Thus, we want to look at the performance in terms of both fairness and throughput maximization. In particular, we compare the baseline MIMO with three algorithms for picking concurrent clients. All three algorithms pick the packet at the head of the queue for transmission; however, they differ in the choice of which other packets to concurrently transmit with it. The first algorithm is a *brute force* search that finds the packets in the queue that maximize throughput.[11] The second algorithm, which we refer to as *FIFO*, combines the packets according to their arrival order. The third algorithm is the *best of two choices*, which is explained in §5.2. This is the choice that IAC adopts.

Experiment. We use the testbed in Fig. 11, where there are three APs and 17 active clients. Each client has infinite demands. This ensures that a client's throughput is not limited by its own demands but by how the concurrency algorithm chooses to serve the client. Packets from different clients arrive at the system in random order. Each run involves using the medium for 1000 timeslots, and we repeat a run 10 times to compute the average rate per client. We

[11] Throughput can be estimated without transmitting the packets as follows: $\sum_i log(1 + ||\vec{v}_i^T H_i \vec{w}_i||^2)$, where the sum is over client-AP pairs, H_i is the channel for a pair, and $\vec{v}_i$ and $\vec{w}_i$ are the corresponding encoding and decoding vectors [33].

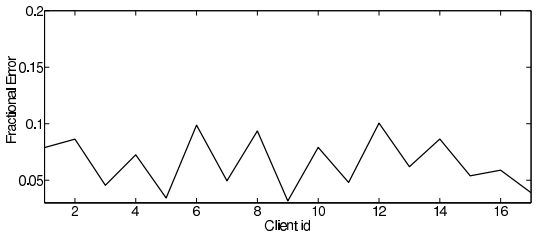


Figure 16—Channel Reciprocity This figure plots the fractional difference between the direct estimate of a downlink channel and its estimate based on reciprocity. The x-axis refers to runs at different clients/locations. The figure shows that estimates based on reciprocity provide a reasonable accuracy and can be used in IAC.

compare four designs: the baseline MIMO, IAC+best-of-two, IAC+brute-force, and IAC+FIFO. For each client, we compute the gain it achieves under a particular concurrency algorithm with respect to the baseline MIMO.

Results. Figs. 15a and 15b show the CDFs of the gains of the three concurrency algorithms with respect to MIMO, both on the uplink and downlink. The figures reveal the following findings:

- All three approaches for choosing concurrent packets provide a significant gain over the baseline MIMO. The average gain on uplink is: 2.32x for the brute force approach, 1.9x for the FIFO approach, and 2.08x for the best-of-two approach. Similarly, on the downlink, the average gain is: 1.58x for the brute force approach, 1.23x for the FIFO approach, and 1.52x for the best-of-two approach. (Note that while IAC's analytical gain is bounded by 2x, the practical gain can be larger because it includes power and diversity gains and also depends on how the concurrency algorithm arbitrates the medium between clients.)
- The three concurrency algorithms differ widely with respect to fairness. In particular, the brute-force approach is significantly unfair. A few clients get humongous throughput, while many clients have a gain smaller than 1, i.e., their rates are better in the baseline MIMO. The other schemes have a better fairness, with the best-of-two approach having the best fairness-throughput tradeoff.
- Thus, IAC's concurrency algorithm that employs the best-of-two approach achieves both fairness and efficiency. It provides an average rate increase of 2.08x on the uplink and 1.52x on the downlink, furthermore all clients see gains over the baseline MIMO.

8.4 Channel Reciprocity

Finally, we check whether channel estimates based on reciprocity are accurate enough to be used in IAC.

Experiment. We take 17 random client-AP pairs from the testbed, and measure their uplink and downlink channels. We compute the calibration matrices according to Eq. 7. For each pair, we then fix the AP and move the client. This causes the uplink and downlink channels to change (but the calibration matrices stay the same.) We now make the AP measure the uplink channel, H^u, and multiply it by the calibration matrices to estimate the downlink channel $H^d_{reciprocity}$.

We compare this estimate with the downlink channel as estimated at the client, H^d_{true}. We compute the fractional error in the AP's estimate of the downlink channel based on reciprocity as $Err = \frac{||H^d_{true} - H^d_{reciprocity}||}{||H^d||}$. We repeat the experiment 5 times for each client, where each run is done in a new location. For each of the 17 client-AP pairs, we plot the average fractional error in Fig. 16.

Results. The figure shows that reciprocity holds to a large extent. The fractional error between the actual downlink channel and the estimate based on reciprocity stays small. Note that since the client changed location between the estimation of the calibration matrices and their later application to estimate the downlink channel, reciprocity is reasonably accurate despite client movement.

9 Conclusion

This paper introduces interference alignment and cancellation (IAC), a new technique that allows MIMO LANs to support a larger number of concurrent transmissions. We analytically show that IAC doubles the throughput of uplink and almost doubles the throughput on the downlink, for a large number of antennas. Our wireless experiments show that, on a testbed of 20 wireless nodes, each equipped with two antennas, IAC provides a 2x gain on the uplink and 1.5x gain on the downlink.

References

[1] Antenna selection and RF processing for MIMO systems. *IEEE 802.11-04/0713r0*, 2004.

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

[3] D. Bliss, K. Forsythe, and A. Chan. Mimo Wireless Communications. *Lincoln Laboratory Journal*, 2005.

[4] D. Bliss, P. Parker, and A. Margetts. Simultaneous Transmission and Reception for improved wireless network performance. In *SSP*, 2007.

[5] V. Cadambe and S. Jafar. Interference Alignment and the Degrees of Freedom for the K User Interference Channel. In *Trans. on Information Theory*, 2008.

[6] P. Castoldi. *Multiuser Detection in CDMA Mobile Terminals*. Artech house Publishers, 2002.

[7] R. Chandra, P. Bahl, and P. Bahl. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. In *INFOCOM*, 2004.

[8] S. Changho and D. Tse. Interference Alignment for Cellular Networks. In *Allerton*, 2008.

[9] A. E. Gamal and T. Cover. Multiple user information theory. In *Trans. on information theory*, 1980.

[10] M. Gast. *802.11 Wireless Networks*. O'Reilly, 2005.

[11] D. Gesbert, M. Kountouris, R. W. Heath, C. Chae, and T. Salzer. Shifting the MIMO Paradigm: From Single User to Multiuser Communications. In *Sig. Proc. Mag.*, 2007.

[12] S. Gollakota and D. Katabi. ZigZag Decoding: Combating Hidden Terminals in Wireless Networks. In *Sigcomm*, 2008.

[13] T. Gou and S. Jafar. Degrees of Freedom of the K user MxN MIMO Interference channel. In *arxiv:0809.0099*, 2008.

[14] M. Guillaud, D. Slock, and R. Knopp. A practical method for wireless channel reciprocity exploitation throught relative calibration. In *Sig. Process. and Apps*, 2005.

[15] R. Gummadi and H. Balakrishnan. Wireless Networks should Spread Spectrum Based on Demands. In *Hotnets*, 2008.

[16] R. Gummadi, R. Patra, H. Balakrishnan, and E. Brewer. Interference Avoidance and Control. In *Hotnets*, 2008.

[17] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: Interference Cancellation for wireless LANs. In *ACM Mobicom*, 2008.

[18] J. Hou, J. Smee, H. D. Pfister, and S. Tomasin. Implementing Interference Cancellation to Increase the EV-DO Rev A Reverse Link Capacity. *IEEE Communication Magazine*, 2006.

[19] C. Huang and S. Jafar. Degrees of Freedom of the MIMO Interference Channel with Cooperation and Cognition. In *arxiv: 0803.1733*, 2008.

[20] S. Jafar and S. Shamai. Degrees of Freedom of the MIMO X Channel. In *Trans in Info. Theory*, 2008.

[21] S. Jakubczak, D. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan. Link-alike: Using Wireless to Share Network Resources in a Neighborhood. In *MC2R*, 2008.

[22] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi. FATVAP: Aggregating AP BackHaul Bandwidth. In *NSDI*, 2008.

[23] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. Dissertation, UCB, 1991.

[24] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *Mobicom*, 2005.

[25] T. Moscribroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan. Load-Aware Spectrum Distribution in Wireless LANs. In *ICNP*, 2008.

[26] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing High Performance Enterprise Wi-Fi Networks. In *NSDI*, 2008.

[27] C. Qiu, C. Zhou, G. Nan, and J. Zhang. Time Reversal with MISO for ultra-wideband Communications: Experimental Results. In *Letters on Antennas and Propogation*, 2006.

[28] C. Qiu, C. Zhou, J. Zhang, and G. Nan. Channel reciprocity and time-reversed propagation for ultra-wideband communications. In *Symp. on Antennas and Propagation*, 2007.

[29] J. Redi, B. Watson, R. Ramanathan, and P. Basu. Design and implementation of a MIMO MAC protocol

for ad hoc networking. *Proc. SPIE*, 2006.
[30] K. Sundaresan and R. Sivakumar. Routing in Ad-Hoc Networks with MIMO Links. In *ICNP*, 2005.
[31] K. Sundaresan and R. Sivakumar. Cooperating with Smartness: Using Heterogeneous Smart Antennas in Ad-Hoc Networks. In *Infocom*, 2007.
[32] K. Sundaresan, R. Sivakumar, M. A. Ingram, and T.-Y. Chang. A Fair Medium Access Control Protocol for Ad-hoc Networks with MIMO Links. In *Infocom*, 2004.
[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 channel and uplink-downlink duality. In *Trans. on information theory*, 2003.
[35] G. Woo, P. Kheradpour, and D. Katabi. Beyond the Bits: Cooperative Packet Recovery Using PHY Information. In *ACM MobiCom*, 2007.

APPENDIX

A. Proof of Lemma 7.1 : Say we have N clients, each with M antennas and client i decodes $n_i > 0$ concurrent packets. So, the total number of concurrent packets is $T = \sum_{i=1}^{N} n_i$. Since client i decodes n_i packets, the remaining $T - n_i > 0$ packets at the client must be aligned in a $M - n_i > 0$ dimensional space resulting in $T - M$ equations. To be consistent, the total number of equations, $N(T-M)$, should be less than the total number of unknown vectors, T, i.e., $N(T-M) \leq T$. Also since $M - n_i > 0$ and $T = \sum_{i=1}^{N} n_i$, we have $T \leq N(M-1)$. Thus, T is bounded by $min\{N(M-1), \lfloor \frac{N}{N-1} M \rfloor\}$. Now, since one can also serve to only $N' < N$ concurrent clients at a time, the number of concurrent packets is upper bounded by $\max_{N' \leq N} \min\{N'(M-1), \lfloor \frac{N'}{N'-1} M \rfloor\}$. When $M = 2$, this bound is $\lfloor \frac{3}{2} M \rfloor$ which occurs at $N' = 3$ and is achieved by the system in Fig 6. When the number of antennas is more than two, the maximum is $2M-2$ and occurs at $N' = 2$. To achieve this, IAC requires $M-1$ APs and two clients. Specifically, similar to Fig 9, each AP transmits two packets, one to each client. At each client, all undesired packets are aligned along one dimension, allowing each client to decode $M-1$ concurrent packets, delivering a total of $2M-2$ packets.

B. Proof of Lemma 7.2 : The following scheme achieves $2M$ streams in a system with three APs each with M antennas. Let $x_1, x_2, \cdots, x_{2M}$ be the 2M streams. At the first AP, streams $x_2, \cdots, x_{2M}$ are aligned in a $M-1$ dimension space which allows x_1 to be decoded. At the second AP, streams $x_{M+1}, \cdots, x_{2M}$ are aligned on one dimension, allowing for streams $x_2, \cdots, x_M$ to be decoded. Finally at the third AP, after cancellation, we are left with the M streams $x_{M+1}, \cdots, x_{2M}$ which can be decoded. Indeed, the variables $v_2, \cdots, v_M$ are involved only in one equation each (for the alignment at AP1) and are free, while the last M vectors, $v_{M+1}, \cdots, v_{2M}$, are involved in a total of M equations (1 equation at AP1 and M-1 at AP2). As long as the channel matrices are random, with probability approaching one, the latter M equations provides a solution for $v_{M+1}, \cdots v_{2M}$. Now using the first M equations for the vectors, $v_2, \cdots, v_M$, we can determine them and hence deliver $2M$ concurrent packets.