iBeam: Intelligent Client-Side Multi-User Beamforming in Wireless Networks

Yongjiu Du\(^1\), Ehsan Aryafar\(^2\), Joseph Camp\(^1\), and Mung Chiang\(^3\)
\(^1\)EE Department, Southern Methodist University, Dallas, TX, USA
\(^2\)Intel Labs, Santa Clara, CA, USA
\(^3\)EE Department, Princeton University, Princeton, NJ, USA

Abstract—Frequently, client-side wireless devices have a view of multiple WiFi access points, whether from open residential and commercial networks, corporate networks, or mesh networks. Given the increasing number of radios and antennas in today's wireless devices, residual capacity from these multiple APs could be leveraged if client devices communicate with multiple APs simultaneously. In this paper, we exploit multi-user multi-input multi-output (MU-MIMO) technology to improve throughput and reliability in both directions of a wireless connection. For uplink, we use multi-user beamforming to enable the client devices to send multiple data streams to multiple APs simultaneously. For downlink, we leverage interference nulling technology to allow the client devices to decode parallel packets from multiple APs. This iBeam system requires no changes to existing APs or backhaul networks and is compatible with the IEEE 802.11 standards. We experimentally evaluate iBeam and show significant throughput improvements over both single-AP connections and multi-AP connections in a time division mode. The client's reliability and stability are also significantly improved due to the multi-AP diversity gain.

Index Terms—Multi-user; Multi-AP; Beamforming; Interference Nulling; Channel Reciprocity; WiFi Networks; Rate Adaptation;

I. INTRODUCTION

Wireless devices have recently offered an increasing number of radios and antennas, which have been leveraged in recent work and WiFi standardization efforts for improved performance from access points to clients [1], [2]. However, despite similar hardware trends for client-side devices, the multiple radios and antennas have yet to be fully leveraged to improve the uplink or downlink performance. In particular, the radio and antenna growth has enabled multi-input multi-output (MIMO) technologies to be leveraged from APs to clients, thereby improving capacity. Beamforming is one such MIMO technology which can improve the received signal strength of the intended user and reduce interference to the unintended users, as has been studied recently [3], [4]. Moreover, multi-user beamforming (MU-beamforming) can achieve additional throughput improvement by properly setting the steering matrix of the transmitting links to enable parallel data transmissions [1], [5]. Another key MIMO technology is interference nulling and cancellation, which allows a receiver to decode multiple parallel streams from different transmitters [6]–[8].

Other work has made use of users frequently being in range of multiple access points with a plethora of available networks in homes or businesses or larger corporate or mesh networks. In [9], for example, the authors introduced a protocol that enables the client device to aggregate the backhaul bandwidth of multiple APs that can be seen by the client, in a time division mode. WiFi, proposed in [10], opportunistically exploits the diversity of multiple APs to minimize disruptions and supports interactive applications for client devices. Given the increasing number of radios in today's wireless devices, residual capacity from these multiple APs could be further exploited if client devices communicate with multiple APs simultaneously as opposed to each AP being multiplexed over time. With the aforementioned MIMO technologies, these multi-AP approaches could be vastly improved to provide increased throughput and reliability. Namely, instead of communicating to multiple APs in a time division mode, with MIMO technology, the client device can maintain connections to multiple APs simultaneously. Adya et al. leverage the simultaneous use of multiple wireless network cards on non-overlapping frequency channels to communicate with multiple APs, achieving significantly improved throughput and latency [11]. However, this scheme does not leverage the efficiency of MIMO and occupies multiple channels, resulting in a large spectral footprint.

Motivated by the high cost of spectrum in licensed bands and high usage in unlicensed bands, we propose iBeam, a wireless system that uses a single channel for multi-AP communication, which can be easily scaled by additional channel availability. In iBeam, the client device can transmit parallel data streams to and receive multiple data streams from multiple WiFi APs simultaneously. We take advantage of channel reciprocity and passive overhearing of control and broadcasting packets transmitted by APs, to avoid incurring channel-state-information (CSI) overhead by the client, thereby improving system efficiency. For MU-beamforming system, power allocation among all the selected beamformees is challenging, especially when time-varying channel and rate adaptation are considered. For the downlink, we use interference nulling technology to enable multiple APs transmitting multiple streams to the client simultaneously. We implement it on the WARP FPGA platform [12] and evaluate it over the air. We experimentally show that iBeam achieves significant improvement in terms of both throughput and reliability. With just 2 antennas on the client device, we achieve an average improvement of 50 percent for the uplink direction and 80 percent for the downlink direction. Also, with multiple-AP connections and smooth handoff, there is almost no outage for the connection.

While there is no shortage of works on advanced physical layer MIMO techniques (e.g., [1], [8]), our goal here is to design and experimentally evaluate a system which leverages such techniques from the point of view of the client to
improve throughput, reliability, and stability. This is in contrast to and complementary with other works which improve the access point performance [6], [13]. In particular, the main contributions of our work are as follows:

- We propose iBeam, a system on client devices to communicate with multiple APs simultaneously, without changing the network infrastructure, access point protocols, or WiFi standard.
- We implement iBeam on a hardware platform by exploiting multi-user beamforming to enable the client device to send multiple uplink data streams to multiple APs simultaneously. For downlink, we use interference nulling technology to allow the client device to decode parallel packets from multiple APs.
- We experimentally evaluate this iBeam system and show significant throughput improvements over single-AP connections or multi-AP connections in a time division mode. The results show an average throughput improvement of 50 percent and 80 percent, for uplink and downlink, respectively.
- We also experimentally analyze the stability and reliability of the iBeam system. In particular, with mobile handoff, we show that iBeam is almost independent of client speed, whereas more traditional techniques suffer severe degradation and long periods of disconnection with user mobility.

The rest of this paper is organized as follows. Section II introduces the iBeam system. In Section III, we present the uplink framework for iBeam, including multi-user beamforming technology, and the method to select the optimal subset of APs for which the client device should connect. We describe the iBeam downlink framework in Section IV. Experimental settings and results are shown in Section V. Then, related work is provided in Section VI. Finally, we conclude in Section VII.

II. iBEAM SYSTEM MODEL

In this section, we present the system model for iBeam, which enables a client device to maintain multiple WiFi access point connections on the same channel, as shown in Figure 1. The client device is equipped with multiple antennas, which are used to connect to multiple APs. In doing so, there are three distinct advantages over a traditional MIMO scheme: (i) clients can remain connected in mobile scenarios via seamless handoffs, (ii) MIMO gains are greater with increased antenna separation on the senders and receivers (as opposed to both having co-located antennas), and (iii) backhaul bandwidth can be aggregated to achieve the capacity of the wireless network. In this WiFi network, the client device is assumed to be able to see several APs operating on the same channel.

For uplink, the client device first selects an optimal subset of APs and then transmits multiple data streams to the selected APs to improve the capacity. For downlink, the associated APs are potentially able to transmit packets simultaneously to the client device, if there are packets in the buffer of each AP. The number of APs that the client device can select does not exceed the number of antennas on the client device. Because of the connections to multiple access points, the handoff is smooth when the client device moves out of the coverage of an AP and moves into the coverage of another AP. The client device can maintain at least one connection to an AP, while making handoff between other APs. Taking the case in Figure 1 as an example, the mobile client is initially connected to AP 5 and AP 8. When the client device moves towards AP 6, the channel between AP 8 and the client device deteriorates and the channel between AP 6 and the client device improves. The client devices then initiates a handoff from AP 8 to AP 6, while always maintaining the connection to AP 5.

In reality, the APs may be densely planned in a large population area, or sparse in remote areas. As a result, the APs may not be able to sense one another. According to carrier sense multiple access (CSMA), only one AP is able to win the time slot and transmit if the APs can sense each other. In the case that the APs cannot hear each other and the client device can see multiple APs, multiple APs can transmit to the client device simultaneously. The client device can decode the parallel signals from the APs, leveraging interference nulling technology.

For uplink, the client device uses MU-beamforming with a zero-forcing method to transmit multiple beams, with different packets on the beams. Zero-forcing eliminates the interference between the APs, which results in less complexity since the beamforming and power allocation algorithm are independent of one another. For iBeam, the client device can communicate to each AP with a separate beam. It is also possible for an iBeam client device to maintain a two-stream connection to one AP and one-stream connections to other APs. In other words, the number of streams between the two APs can be heterogeneous.

The iBeam system requires no additional coordination between the multiple APs in use at a given time (i.e., the multi-AP intelligence comes from the client-side device) and needs no changes to each AP’s protocols (e.g., message passing) or standards (e.g., iBeam is compatible with IEEE 802.11). The traffic management when using multiple APs is well discussed in prior works [9], [11], [14], and we can abstract the client device to appear as multiple virtual interfaces, for both the upper layer applications and the APs. We will then configure each virtual card to connect to a different AP. iBeam also requires the client device to encode/decode multiple spatial
III. iBEAM UPLINK

For the uplink direction of the iBeam system, we apply multi-user beamforming on client devices to transmit parallel data streams to a set of available APs. To avoid injecting additional overhead, we use control and broadcasting packets from the AP (e.g., beacons) to measure the downlink channel and leverage channel reciprocity to estimate the uplink channel. In Section V, we experimentally evaluate the channel reciprocity assumption with respect to channel quality. According to the channel gains, the set of APs is chosen in such a way to obtain the optimal capacity. When the client device moves, the channel between the client device and the connected APs will be altered. The client device then needs to update the channel information and adaptively change the beamforming parameters to maintain the zero-forcing conditions. Furthermore, the selected optimal subset of connected APs may also change. As a result, the client device will initiate mobile handoffs between APs to maintain the optimal performance.

A. Multi-User Beamforming

In this section, we describe multi-user beamforming (MU-Beamforming), which is a key uplink technology in our iBeam system. Assume one client device with $M$ antennas beamforms to $N$ APs, and each AP has one antenna. For MU-Beamforming, we can express the received signal at AP $i$ as:

$$y_i = h_i \times [q_1, q_2, ..., q_N] \times x + n$$
$$= (h_i, q_i) x_i + \sum_{j \neq i} h_i, q_j x_j + n$$
$$= (h_i, q_i) \sqrt{P_i} s_i + \sum_{j \neq i} h_i, q_j \sqrt{P_j} s_j + n$$  \hspace{1cm} (1)

Here, $x = [x_1, x_2, ..., x_N]^T$ is the transmit signal vector for all $N$ receivers. $h_i = [h_{i,1}, h_{i,2}, ..., h_{i,M}]$ is the channel vector from the beamformer to beamformee $i$. $q_i = [q_{i,1}, q_{i,2}, ..., q_{i,M}]^T$ is a steering vector from the beamformer to beamformee $i$. $n$ is a vector of additive noise and may include interference from other networks. $s_i$ is the normalized transmitted signal with transmit power $P_i$.

In Equation (1), the first and second terms represent the intended transmission and interference from other APs, respectively. Each receiver decodes the transmitted packet by treating both the interference and noise as an additive Gaussian noise. Then, the achieved sum rate is:

$$R = \max_{q_n, P_n} \sum_{n=1}^{N} \log(1 + \frac{P_n |h_n, q_n|^2}{\sum_{j=1, j \neq n}^{N} P_j |h_n, q_j|^2 + \sigma^2})$$  \hspace{1cm} (2)

subject to $\sum_{n=1}^{N} |q_n|^2 P_n \leq P$.

To determine the optimal $q_n$ and $P_n$ to achieve the maximum rate is highly complex, especially when there is a large number of receivers. Zero-forcing is a sub-optimal but comparatively simpler solution for this problem [5], [15], [16]. The key idea is to properly select $q_j$ to make $h_i, q_j = 0$ for all $j \neq i$.

Assume the client device transmits packets to a subset of $S$ APs out of $N$. Let $H(S)$ and $Q(S)$ denote the corresponding sub-matrices of $H = [h_1^T \cdots h_N^T]^T$ and $Q = [q_1 \cdots q_N]^T$, respectively. The solution of $Q(S)$ to obtain zero-forcing is the pseudo inverse of $H(S)$:

$$Q(S) = H(S)^* (H(S) H(S)^*)^{-1}$$  \hspace{1cm} (3)

Then, we can choose $P_j$ based on the water-filling algorithm to maximize the total rate, or use other power allocation algorithms with consideration of fairness between the APs.

The client device is able to control the packet size for each AP to allow transmissions to different APs to end at the same time, which increases spatial reuse, and allows the ACKs mechanism to be simple. After successfully receiving the packets, the APs send ACKs back to the client device. The client device then decodes the ACKs simultaneously, leveraging the interference nulling technology described in Section IV.

B. Semi-Orthogonal AP Selection

As discussed above, we nullify interference from the transmissions to different APs by forcing $h_i, q_j = 0$, for $k \neq j$. In order to maximize the transmit power efficiency for AP $j$, a preferred steering vector should satisfy $q_j = h_j^*/||h_j||$. Therefore, when $h_k, h_j^* = 0$, for $k \neq j$, both conditions are satisfied, and both the beamforming gain and the multi-user diversity gain can be obtained. This means that, for different APs, if their channel vectors to the client device are orthogonal to each other, we obtain the optimal gain. However, in practice, it is hard to find the channel vectors that are absolutely orthogonal to each other, especially for small number of APs. A sub-optimal but feasible solution is to find the subset of APs that have the best orthogonality, which forms a semi-orthogonal subset of APs.

For ease of presentation, each client device is equipped with two antennas, which allows it to communicate with two APs simultaneously with MU-beamforming. Upon entering the network coverage area, the client device detects all the potential APs for connection and obtains the channel state information to each AP. Among the detected APs, it finds the two optimal APs by jointly considering the channel gains and the channel orthogonality. To do so, we propose (and will implement and evaluate) the following greedy algorithm:

1) For each AP seen by the client device, calculate the norm of the channel gain $\rho_i = ||h_i||$ and select the AP that maximizes $\rho_i$.

2) For each of the remaining APs, find the value of $\alpha_i = \frac{h_i, h_j^*}{||h_i|| \cdot ||h_j||}$ and choose the AP with the value of less than or equal to a constant $\alpha$. If none of the APs satisfy this condition, then the client device only communicates to the AP selected in step 1.

3) If at least one AP satisfies the requirement in step 2, we calculate $\eta_j = h_j - \frac{h_j, h_j^*}{||h_j||^2} h_j$ and choose the AP with the maximum $|\eta_j|$ as the secondary AP.

This method easily scales to more antennas and more simultaneous APs for even greater levels of performance.
device, respectively. Then we have:

\[ y_1 = h_{11}x_1 + h_{12}x_2 + n_1 \]
\[ y_2 = h_{21}x_1 + h_{22}x_2 + n_2 \]  

(4)

C. AP Adaptation

When the client device moves, the channel gains and orthogonality between the client device and the APs are altered. In order to maintain a zero-forcing transmission, the client device needs to adapt the beamforming parameters to the changing channel conditions. Moreover, the changing channel conditions may result in a different optimal subset of connected APs. The client device may reselect the optimal subset of APs according to the updated channel state information. If the selected subset of APs changes, the client device needs to initiate the handoff process between the disconnecting APs and the newly selected APs, to optimize the system performance. In order to maintain the service quality, there are generally two rules to follow. First, the client device needs to maintain at least one connection to an AP during the handoff between other APs to avoid service disruption. Second, the handoff should not be too sensitive to the channel quality, to avoid handing off too frequently. We can apply existing handoff methods as described in [17] to successfully adapt APs.

IV. iBEAM DOWNLINK

A device with multiple antennas can potentially decode multiple parallel packets simultaneously on the same channel. With the knowledge of CSI, we can apply interference nulling with or without interference cancellation techniques to decode packets simultaneously from different APs. For the case that the packets from different APs do not end at the same time, ACK management from the client device is challenging. Wireless full duplex allows the client device to send an ACK to an AP while still receiving signals from other APs.

A. Interference Nulling

When multiple APs transmit different packets to a client device, the packets from different APs interfere with one another. However, leveraging the multiple antennas on the client device, we can properly remove the interference and decode each of the intended packets. Taking the system shown in Figure 2 as an example, AP 1 and AP 2 transmit two different packets \( x_1 \) and \( x_2 \), respectively, to the client device. The client device is equipped with 2 antennas. Let \( y_1 \) and \( y_2 \) denote the packets received by the two antennas on the client device, respectively. Then we have:

\[ y_1 = h_{11}x_1 + h_{12}x_2 + n_1 \]
\[ y_2 = h_{21}x_1 + h_{22}x_2 + n_2 \]  

(4)

Fig. 2: Two APs transmit packets to the same client device simultaneously.

We can also denote the dual packet reception in matrix form as:

\[ Y = HX + N \]  

(5)

where,

\[ Y = [y_1, y_2]^T \]
\[ X = [x_1, x_2]^T \]
\[ H = [[h_{11}, h_{12}]^T, [h_{21}, h_{22}]^T]^T \]  

(6)

We can remove the interference and obtain \( x_1 \) and \( x_2 \) by multiplying \( H^{-1} \) on both sides of Equation (5) to obtain:

\[ X = H^{-1}Y + H^{-1}N \]  

(7)

This could also be explained as channel vector operation used in [8]. We plot the channel vectors \( h_{v1} = [h_{11}, h_{21}] \) and \( h_{v2} = [h_{12}, h_{22}] \) on the plane spanned by the two antennas on the client device, as shown in Figure 3. In order to decode \( x_1 \), we project \( h_{v1}x_1 \) in the direction that is orthogonal to \( h_{v2} \). This nullifies the interference from \( x_2 \), and we can decode \( x_1 \). Similarly, we can we project \( h_{v2}x_2 \) in the direction that is orthogonal to \( h_{v1} \). This nullifies the interference from \( x_1 \), and we can decode \( x_2 \). Note that because of the projection operation, as discussed in [8], the resulted effective SNR is reduced to \( sin^2(\theta)/SNR_{orig} \), which is also shown in Figure 3.

Here, \( \theta \) is the angle between \( h_{v1} \) and \( h_{v2} \) and can be calculated as:

\[ \cos \theta = \frac{|h_{v1}|^2 |h_{v2}|^2}{|h_{v1}||h_{v2}|} \]  

(8)

If \( h_{v1} \) and \( h_{v2} \) are orthogonal, \( \theta \) will be \( \pi/2 \). As discussed in Section III, we have selected the semi-orthogonal APs. Hence, the effective SNR will not significantly degrade.

B. Interference Cancellation

As discussed above, interference nulling usually results in reduced effective SNR. Another option is to decode the packets from one of the APs by interference nulling. Then, we can reconstruct the transmission and cancel it from the mixed reception to obtain the original packet from another AP. The whole process can be divided into the following steps:

1) Use interference nulling to decode the packet from one of the APs (taking \( x_1 \), for example) and estimate the
sampling frequency offset, carrier frequency offset, and channel impulse response during the decoding.

2) Encode and modulate the decoded packet and reconstruct the received packet according to the channel response, sampling frequency offset, and carrier frequency offset.

3) Remove the reconstructed packet (\(x_1\) for example) from the mixed reception and then decode the remaining packet (\(x_2\)) with no interference.

Ideally, using interference cancellation, we can remove the interference of parallel transmissions to obtain the second packet and achieve better performance than simply nulling the interference from the first AP. However, in practice, it is hard to reconstruct the interfering packet without errors, because of the residual carrier frequency error, residual sampling frequency error, and residual channel estimate errors. The reconstruction errors will be the residual interference. As discussed in [6]–[8], the residual interference is usually greater than the noise level. Thus, using interference cancellation, the resulting effective SNR is less than half of the original SNR. Noting that if we use interference nulling on both packets, due to the channel orthogonality of the selected APs, the resulted SNR usually degrades less than 50 percent (i.e., the resulting effective SNR is not as degraded with interference nulling as opposed to interference cancellation). Additionally, the reconstruction of the packets is far more complex and requires far greater computational/hardware resources than the interference nulling. Furthermore, this will also cause substantial delays because of the decoding and reconstruction process. As a result, we only use interference nulling on simultaneous transmissions in the iBeam system, which could significantly reduce the computational complexity while maintaining high throughput.

C. Compatibility With IEEE 802.11 Standard

In a WiFi access network, an ACK packet is required when the receiver successfully decodes a unicast data packet. Otherwise, the transmitter will treat the data packet as a transmission failure and re-transmit the same packet. For a corporate network, the APs could be cooperative, either being controlled by a network server or through exchanging control information between one another. In this case, the APs selected by the client device may not necessarily start their transmissions at the same time, but the transmitted packets can be synchronized to end at the same time. Then, the client device can send ACKs to all APs simultaneously. We call this synchronized ACKs.

However, for distributed APs that are not able to be synchronized, their transmissions probabilistically do not end at the same time. After successfully decoding a packet from one of the APs, the mobile client is required to send an ACK back to that AP. However, for half-duplex wireless devices, receiving the packets from other APs that end later would prevent the ACK packets from being sent back to the AP that had the packet end first. Full duplex wireless communication is able to solve this ACKs problem. With full duplex, the client devices can actually transmit ACK packets while receiving other packets from APs. We call this asynchronized ACKs.

Wireless full duplex has been well studied and practical experiments have shown that it is ready to be used in wireless communications [18], [19].

V. IMPLEMENTATION AND EXPERIMENTS

In this section, we evaluate the iBeam system, including the feasibility of our channel reciprocity assumption and the uplink and downlink portions of the iBeam system. The analysis is performed on WARP [12], a software defined radio platform with extensive over-the-air experimentation.

A. Experiment Setup

We implement and evaluate iBeam on WARP, a software defined radio platform which enables users to implement the physical layer and MAC layer functions in MATLAB in the form of software code. Then, the coded and modulated data samples are transferred to the WARP board via an Ethernet cable. WARP is then triggered to transmit data samples over the air. The receiver samples the received signal over the air and then transfers the raw samples to the PC, where the receiver also leverages MATLAB to demodulate and decode the sampled signal to obtain the packets that the transmitter sends. We use one WARP board as the client device, which is equipped with two antennas. To show the practical application of our system, we choose antennas with small size and mount them at approximately 10 cm apart on a plastic board. The spacing emulates an area of a typical commercial smartphone, as shown in Figure 4. We implemented an OFDM scheme with 64 subcarriers, similar to IEEE 802.11 devices. We use a 10 MHz bandwidth and run the experiments at 2484 MHz carrier frequency. In our system, we can choose between the same 8 bit rates as present in IEEE 802.11a/g by setting the code rates and modulation orders, accordingly. For uplink, the client device chooses different packet sizes for different bit rates to synchronize different beams. For downlink, because WARP system does not support wireless full duplex, we use synchronized ACKs in our implementation and evaluation. We control the packets transmission from different APs to start and end at the same time. With the same frame slot, we transmit \{960, 1440, 1920, 2880, 3840, 5760, 7680, 8640\} bits for each bit rate, respectively.

Rate adaptation significantly improves the spectrum efficiency in IEEE 802.11a/g/n systems and has been well studied [20], [21]. A MU-beamforming system also requires rate adaptation strategies and algorithms to achieve greater levels of spectrum efficiency. For multi-user systems, rate adaptation must be jointly considered with power allocation and channel variations, which is more challenging than in single-user systems.

For MU-Beamforming, power allocation has multiple rules to follow. For example, an aggressive metric is to maximize the total throughput. Some other metrics could be throughput fairness or transmission time fairness between APs. However, when the user connects to multiple APs via iBeam, the fairness between APs is not as critical from the client’s point of view. Thus, the maximum total throughput that the client experiences could be achieved according to the water-filling algorithm.
A channel reciprocity calibration has been proposed and phase offset, and to recover the reciprocity of the overall channel no longer reciprocal. Sufficient levels of training is regarded as part of the channel, which makes the overall components through which the signal passes could also be transmitted between the transmitter and the receiver, however, the digital and analog only applies in wireless channels. For the channel between the two stations A and B, we calculate $H_{AB}$ as the estimated channel state information. For protocols with CSI feedback defined (e.g., IEEE 802.11n, IEEE 802.11ac), both implicit CSI feedback and explicit CSI feedback can be used to get the estimated channel parameters and construct the steering matrix for the beamformer. In implicit feedback mode, the beamformer relies on channel reciprocity in the time division duplex channel to estimate the channel from the beamformer to the beamformee. In iBeam, we leverage the downlink channel state information, which is the predominant rate adaptation algorithm implemented in commercial APs.

### B. Channel Reciprocity Evaluation

For iBeam, the received signal strength (RSS) depends on both the channel variations and the power allocation. The client device can predict the received signal strength after getting the steering matrix and the power allocation. As a result, the RSS-based rate adaptation is the most straightforward method for uplink (e.g., the one we used in [22]). According to the received signal strength, an optimal bit rate can be selected to both ensure the packet delivery and maximize the spectrum efficiency. For downlink, the APs can use packet-loss-based rate adaptation algorithm, which is the predominant rate adaptation algorithm implemented in commercial APs.

For iBeam, the received signal strength (RSS) depends on both the channel variations and the power allocation. The client device can predict the received signal strength after getting the steering matrix and the power allocation. As a result, the RSS-based rate adaptation is the most straightforward method for uplink (e.g., the one we used in [22]). According to the received signal strength, an optimal bit rate can be selected to both ensure the packet delivery and maximize the spectrum efficiency. For downlink, the APs can use packet-loss-based rate adaptation algorithm, which is the predominant rate adaptation algorithm implemented in commercial APs.

The residual calibration errors affect each packet transmission. When the channel quality is poor during calibration, the result would be with more errors, resulting in poor performance for the following packet transmission. A simple way for calibration with higher accuracy is to perform the calibration only when the received SNR is higher than a threshold. Each client device needs to be precalibrated by an AP with CSI feedback, this could be done during manufacturing or by the user during daily use.

We use two WARP boards sending signals back and forth to evaluate the performance and accuracy of the CSI estimation based on channel reciprocity. The PC connected to both WARP boards controls one board as the transmitter and the other as a receiver. We then alternate the sender and receiver every other transmission, as shown in Figure 5. We change the transmit power to see how the received signal strength affects the channel estimation errors due to the additive Gaussian noise in the system.

For iBeam, the received signal strength (RSS) depends on both the channel variations and the power allocation. The client device can predict the received signal strength after getting the steering matrix and the power allocation. As a result, the RSS-based rate adaptation is the most straightforward method for uplink (e.g., the one we used in [22]). According to the received signal strength, an optimal bit rate can be selected to both ensure the packet delivery and maximize the spectrum efficiency. For downlink, the APs can use packet-loss-based rate adaptation algorithm, which is the predominant rate adaptation algorithm implemented in commercial APs.

The residual calibration errors affect each packet transmission. When the channel quality is poor during calibration, the result would be with more errors, resulting in poor performance for the following packet transmission. A simple way for calibration with higher accuracy is to perform the calibration only when the received SNR is higher than a threshold. Each client device needs to be precalibrated by an AP with CSI feedback, this could be done during manufacturing or by the user during daily use.

We use two WARP boards sending signals back and forth to evaluate the performance and accuracy of the CSI estimation based on channel reciprocity. The PC connected to both WARP boards controls one board as the transmitter and the other as a receiver. We then alternate the sender and receiver every other transmission, as shown in Figure 5. We change the transmit power to see how the received signal strength affects the channel estimation errors due to the additive Gaussian noise in the system.

For iBeam, the received signal strength (RSS) depends on both the channel variations and the power allocation. The client device can predict the received signal strength after getting the steering matrix and the power allocation. As a result, the RSS-based rate adaptation is the most straightforward method for uplink (e.g., the one we used in [22]). According to the received signal strength, an optimal bit rate can be selected to both ensure the packet delivery and maximize the spectrum efficiency. For downlink, the APs can use packet-loss-based rate adaptation algorithm, which is the predominant rate adaptation algorithm implemented in commercial APs.

The residual calibration errors affect each packet transmission. When the channel quality is poor during calibration, the result would be with more errors, resulting in poor performance for the following packet transmission. A simple way for calibration with higher accuracy is to perform the calibration only when the received SNR is higher than a threshold. Each client device needs to be precalibrated by an AP with CSI feedback, this could be done during manufacturing or by the user during daily use.

We use two WARP boards sending signals back and forth to evaluate the performance and accuracy of the CSI estimation based on channel reciprocity. The PC connected to both WARP boards controls one board as the transmitter and the other as a receiver. We then alternate the sender and receiver every other transmission, as shown in Figure 5. We change the transmit power to see how the received signal strength affects the channel estimation errors due to the additive Gaussian noise in the system.

For iBeam, the received signal strength (RSS) depends on both the channel variations and the power allocation. The client device can predict the received signal strength after getting the steering matrix and the power allocation. As a result, the RSS-based rate adaptation is the most straightforward method for uplink (e.g., the one we used in [22]). According to the received signal strength, an optimal bit rate can be selected to both ensure the packet delivery and maximize the spectrum efficiency. For downlink, the APs can use packet-loss-based rate adaptation algorithm, which is the predominant rate adaptation algorithm implemented in commercial APs.
the signals from different transmit antennas can always be
signal strength at both APs. However, for SU-beamforming,
directly point to the intended AP, leading to a low received
the zero-forcing requirement, the resulted beams could not
of the 2 APs being less orthogonal. Here, in order to satisfy
Figure 8a and 8b where the throughput of MU-beamforming is
percent in terms of throughput. Nonetheless, there are cases in
beamforming mode outperforms the SISO mode more than 50
mode more than 30 percent in terms of throughput. The MU-
the client device, which randomly selects locations within the
gain, which we call SISO. In the second scheme, we force the
device. For each location, we test the throughput of three
schemes. In the first scheme, we force the client device to
use one antenna to connect to the AP with the best channel
gain, which we call SISO. In the second scheme, we force the
client device connect to the AP with the best channel gain with
single-user beamforming, which we call SU-Beamforming. In
the third scheme, we force the client device connect to multiple
APs with MU-beamforming. For all three schemes, we use
the same total transmit power from the client device. The
APs with MU-beamforming. For all three schemes, we use
third scheme, we force the client device connect to multiple
locations for the client device. At each location, we measure
the throughput of iBeam and single-AP connection. Then,
we plot the throughput in Figure 8c. We can see that, with
two-AP transmission, the downlink throughput is improved as
much as 80 percent. However, similar to the uplink direction,
we can see cases where the throughput with multiple APs
connected is less than the SISO scheme. This also results from
the non-orthogonality of the channels from different APs to
the client device. As we discussed in Section IV, the signal
projection causes signal strength degradation from each AP. If
the channel gain of second AP is far worse than the first AP,
the capacity gain from the second AP may not compensate for
the projection loss of the first AP, leading to a lower throughput
than the SISO scheme.

E. Handoff Evaluation

When the client device moves, the channel gains to each
AP change with time. For a single-AP system, when the
client device moves out of the range of the current connected
AP, handoff occurs. However, the client is disconnected until
the client device is associated with another AP. This handoff
process usually takes as long as hundreds of milliseconds to
even several seconds, depending on the handoff algorithms,
implementation, and traffic load of the network [17]. The
frequency of handoff depends on the speed at which the client
device travels. The faster the client devices moves, the more
frequently handoffs occur, leading to an increasingly unstable
connection. In our design, the client device is connected to
two APs at the same time. When there is a handoff for one
of the APs, the other AP is still in service to avoid the outage
typically experienced with mobile handoff.

The traditional handoff process includes channel scanning,
authentication, and re-association as well as any updates
necessary on the wired network. Channel scanning consumes
most of the handoff time because there are potentially tens
of channels to scan. The client devices send probe packets,
listen to feedback packets, and switch to another channel.
Authentication, re-association, and wired network updates take
less than 40 ms in total [17]. For the iBeam system, both the
client devices and the APs work on the same channel. As a
result, the probing delay is significantly reduced. The total handoff delay for iBeam is less than 60 ms (assuming the maximum probe delay is 20 ms, according to [17]).

Although WARPLab transmits signals over the air, it cannot provide real-time interaction with commercial APs due to different inter-frame spacing and insufficient turn-around time. Thus, we create a simple network topology to demonstrate the handoff performance of iBeam versus time-division (TD) multi-AP systems. As shown above, with only 2 antennas on the client device, we obtain about 1.5 times uplink throughput gain and about 1.8 times downlink throughput gain. Assuming that the uplink and downlink offered load are equal, then the average total throughput gain of iBeam is about 1.65 times of the time-division scheme. In our model, we assume the average throughput of the time-division scheme is normalized to 1. Then, the iBeam throughput is normalized to 1.65.

As shown in Figure 9a, the client device moves from location A to C. For the time-division scheme, it connects to AP 1 when it moves from location A to B, and connects to AP 3 when it moves from location B to C. The handoff occurs when the client device is crossing location B. In Figure 9b, for iBeam, the client device connects to AP 1 and AP 2 when it moves from location A to B, and transfers to AP 3 and AP 4 when it moves from location B to C. Note that, for iBeam, the client device first makes handoff between AP 2 and AP 4. After being associated to AP 4, the client device then hands off between AP 1 and AP 3. When the client device is at location A, the received signal strength is $S_A$.

Assuming that location A is the midpoint of AP 1 and AP 2 and location C is the midpoint of AP 3 and AP 4, when the client moves a distance $x$ towards C, the received signal strength is then $S_x = S_A d^2/(4x^2 + d^2)$, with $d$ denoting the distance between AP 1 and AP 2. Then, the instantaneous capacity is $B \log_2(1 + S_x/N)$. We can calculate how the handoff affects the system throughput.

In our simulation, we set $d = 20m$ and we assume the time-division scheme has a handoff time of one second. We calculate the average throughput of both schemes when the client device moves from location A to C, with different speed. We also simulate the instantaneous throughput of both schemes when the client devices moves at a speed of 1 m/s. As shown in Figure 10a and 10b, we simulate how handoff affects the communication for the two schemes. We can see that, because of the small handoff time, the iBeam throughput is largely independent of client speed. However, the throughput of the time-division scheme drops significantly when the speed increases since the portion of outage time is correspondingly increased. Moreover, for iBeam, because there is at least one AP connection at any time, there is no outage during the handoff. However, for the time-division scheme, there is outage for each handoff.

**VI. RELATED WORK**

The majority of the work on multi-user communication has been from the network to the client device. This includes coordinated multipoint to point or network MIMO techniques.
in cellular networks [25], as well as wireless LANs [7], [13], [24]. These schemes require a high-speed central controller that is connected to each AP and antennas in a cellular or enterprise wireless network, which does not exist in mesh deployments or when WiFi APs belong to different service providers. Other work has focused on enhancing the performance of mobile devices through communication with multiple APs that belong to different access technologies [26]–[28], or the same technology but tuned to non-overlapping frequency channels [9], [29]–[31]. Other work has proposed methods that use directional antennas or beamforming techniques at the mobile device [3], [32], [33], but these works are limited to single-AP communication. In contrast, we design and implement a system that leverages the increasing computational and communication resources of client-side devices to provide the high capacity and reliability gains associated with multi-user communication to multiple access points while consuming a small spectral footprint.

VII. CONCLUSION

In this paper, we designed iBeam, an intelligent client-side MIMO framework to leverage simultaneous use of multiple access points. For the uplink traffic, we leveraged MU-beamforming technology to enable the client devices to transmit multiple data streams to multiple APs simultaneously. For downlink, we used interference nulling to enable the client devices to decode parallel transmissions from multiple APs. With MIMO techniques applied to both uplink and downlink directions, we significantly improved the system throughput due to the multi-user and beamforming gains. With only two antennas on the client device with antenna spacing comparable to a smartphone, iBeam achieves an average improvement of 50 percent for the uplink traffic and 80 percent for downlink traffic. This improvement will only increase with the use of more antennas or greater antenna separation, which could be offered by non-mobile clients. For mobile users, the stability and reliability are significantly improved due the multi-user diversity across APs.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their valuable feedback, which improved the paper. This work was supported in part by NSF grants: CNS-1150215, CNS-1318607, and CNS-1320442.

REFERENCES