AirSync: Enabling Distributed Multiuser MIMO with Full Spatial Multiplexing

Horia Vlad Balan
USC
hbalan@usc.edu

Ryan Rogalin
USC
rogalin@usc.edu

Antonios Michaloliakos
USC
michalol@usc.edu

Giuseppe Caire
USC
caire@usc.edu

Konstantinos Psounis
USC
kpsounis@usc.edu

ABSTRACT

The enormous success of advanced wireless devices is pushing the demand for higher wireless data rates. Denser spectrum reuse through the deployment of more access points per square mile has the potential to successfully meet the increasing demand for more bandwidth. In theory, the best approach to density increase is via distributed multiuser MIMO, where several access points are connected to a central server and operate as a large distributed multi-antenna access point, ensuring that all transmitted signal power serves the purpose of data transmission, rather than creating “interference.” In practice, while enterprise networks offer a natural setup in which distributed MIMO might be possible, there are serious implementation difficulties, the primary one being the need to eliminate phase and timing offsets between the jointly coordinated access points.

In this paper we propose AirSync, a novel scheme which provides not only time but also phase synchronization, thus enabling distributed MIMO with full spatial multiplexing gains. AirSync locks the phase of all access points using a common reference broadcasted over the air in conjunction with a Kalman filter which closely tracks the phase drift. We have implemented AirSync as a digital circuit in the FPGA of the WARP radio platform. Our experimental testbed, comprised of two access points and two clients, shows that AirSync is able to achieve phase synchronization within a few degrees, and allows the system to nearly achieve the theoretical optimal multiplexing gain. We also discuss MAC and higher layer aspects of a practical deployment. To the best of our knowledge, AirSync offers the first ever realization of the full multiuser MIMO gain, namely the ability to increase the number of wireless clients linearly with the number of jointly coordinated access points, without reducing the per client rate.

Categories and Subject Descriptors

C.2.2 [**Computer System Organization**]: Computer Communication Networks

General Terms

Design, Experimentation, Performance

Keywords

Wireless, Virtual MIMO, Software Radios, Synchronization

1. INTRODUCTION

The enormous success of advanced wireless devices such as tablets and smartphones is pushing the demand for higher and higher wireless data rates and is causing significant stress to existing networks. While new standards (e.g., 802.11n and 4G) are developed almost every couple of years, novel and more radical approaches to this problem are yet to be tested. The fundamental bottleneck is that wireless bandwidth is simply upper bounded by physical laws, in contrast to wired bandwidth, where putting new fiber on the ground has been the de-facto solution for decades. While advances in network protocols and modulation and coding schemes have managed relatively modest improvements, denser spectrum reuse, that is placing more access points per square mile, has the potential to successfully meet the increasing demand for more bandwidth. However, very dense infrastructure deployments cannot be carefully planned and managed for reasons pertaining to scale and cost. Therefore, the denser the deployment, the larger the interference among different access points. Eventually the system becomes interference-limited and we are back to square one.

In theory, the ultimate answer to this problem is distributed multiuser MIMO (also known as “virtual MIMO”), where several (possibly multi-antenna) access points are connected to central servers and operate as a large distributed multi-antenna base station. When using joint decoding in the uplink and joint precoding in the downlink, all transmitted signal power is useful, as opposed to conventional random access scenarios (e.g., carrier-sense) which waste power through interference. This approach is particularly suited to the case of an enterprise network (e.g., a WLAN covering a conference center, an airport terminal or a university), or to the case of clusters of closely spaced home networks connected to the Internet infrastructure through the same cable bundle.

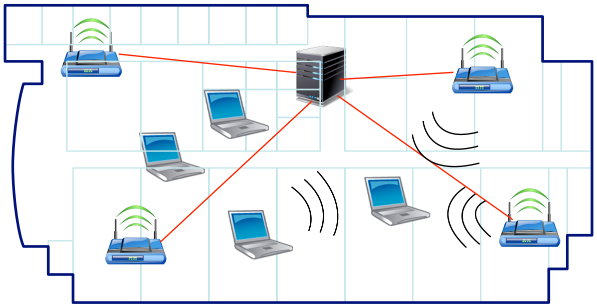

Figure 1: Enterprise Wifi and Distributed MIMO. Multiple access points connected to a central server through Ethernet (red lines) coordinate their transmissions to several clients by using distributed MIMO.

However, distributed multiuser MIMO is regarded today mostly as a theoretical solution because of some serious implementation hurdles, such as the ability to eliminate phase and timing offsets between jointly coordinated access points and the ability to perform efficient joint encoding at a central server linked to the access points through wired links of limited capacity.

We consider a typical enterprise network as illustrated in Figure 1. Since in such networks the wired links connecting the access points are fast enough to allow for efficient joint processing at a server, the major obstacle to achieve the full potential of distributed MIMO gains is eliminating the phase offsets between the different access points. The perceived difficulty of this task has led some researchers to believe that it is practically impossible to achieve full spatial multiplexing in the context of distributed MIMO. In this paper, we present the first (to the best of our knowledge) real-world testbed implementation which achieves the theoretical optimal gain by removing, in real time, the phase offsets between geographically separated access points. We achieve this via *AirSync*.

AirSync is a novel scheme which provides not only time but also phase synchronization between access points. In a nutshell, AirSync locks the phase of all access points using a common reference broadcasted over the air in conjunction with a Kalman filter which closely tracks the phase drift between the different oscillators. We have implemented AirSync as a digital circuit in the FPGA of the WARP radio platform. We have also implemented *Zero-Forcing Beamforming*, a physical layer precoding scheme for multiuser downlink transmission, and investigated the practical requirements of optimal MAC-layer schemes. Finally, we have shown in a testbed consisting of four WARP radios, two acting as access points connected to a central server and two acting as clients, that the theoretical optimal gain of multiuser MIMO is achievable in practice. We argue later in the paper that this result will extend to an increasing number of access points as long as there is enough spatial diversity in the propagation environment. This is something that depends entirely on the richness of the physical channels and has nothing to do with the distributed nature of our MIMO system and AirSync. While recently there have been a number of very interesting and important works in which some of the gains of multiuser MIMO have been shown (see Section 2 for more details) none of these has managed to achieve phase synchronization between remote transmitters and thus they all fall short of the optimal gains in the distributed sender scenario.

In summary, the contributions that we make in this paper are the following:

- We introduce AirSync, the first (to the best of our knowledge) scheme which achieves phase synchronization in a distributed multiuser MIMO setting.
- We implement AirSync as a digital circuit in the FPGA of the WARP platform.
- We showcase in a testbed consisting of 4 WARP radios that, thanks to AirSync, the theoretically optimal spatial multiplexing gain is achievable in practice.
- We discuss practical implementation aspects of the theoretically optimal MAC schemes to be used in conjunction with our distributed MIMO system.

We conclude this introduction by providing a brief outline for the rest of the paper. In Section 2 we discuss in detail related work both on the theoretical side (information theory) and on the practical side (software radio implementations). In Section 3 we use a theoretical approach to show why phase synchronization is needed to achieve the promised gain, and describe, in general terms, AirSync. In Section 4 we present the hardware implementation of AirSync in detail. In Section 5 we present a number of results obtained using our testbed implementation with two access points and two clients. We show results regarding the synchronization accuracy, the beamforming gain, the Zero-Forcing precision and the multiuser multiplexing gain of the system. The following section mentions theoretically optimal MAC schemes and efficient approximations as well as their practical realizations. Finally, Section 7 discusses a number of challenging yet promising topics that we plan to explore in the future, namely the use of rateless codes for flexible dynamic scheduling and implicit rate allocation, as well as possible alternative non-linear multiuser precoding schemes.

2. RELATED WORK

The pioneering papers by Foschini [13] and Telatar [31] have shown that adding multiple antennas both to the transmitter and to the receiver increases the capacity of a point-to-point communication channel. At practical medium-to-high Signal to Noise Ratios (SNRs), this gain manifests as a multiplicative factor equal to the rank of the matrix representing the transfer function between the transmit and the receive antennas. For sufficiently rich propagation scattering, with probability 1 this factor is equal to $\min\{N_t, N_r\}$, where N_t and N_r denote the number of transmit and receive antennas, respectively. The MIMO capacity gain can be interpreted as the implicit ability to create $\min\{N_t, N_r\}$ "parallel" non-interfering channels corresponding to the channel matrix eigenmodes, and it is referred to in the literature as *multiplexing gain*, or as the *degrees of freedom* of the channel. Subsequently, Caire and Shamai [4] have shown that the MIMO broadcast channel, where the transmitter has N_t antennas and serves K clients with N_r antennas each, exhibits an analogous capacity factor increase of $\min\{N_t, KN_r\}$, suggesting that a transmitter with multiple antennas could transmit simultaneously on the same frequency to independent users. Such multiuser communication has two additional requirements. First, precoding of the transmitted data is needed to prevent the different spatial streams from mutually interfering. Second, the transmitter requires accurate knowledge of the channel matrix (channel state information) in order to realize this precoding.

The idea of precoding has spurred research beyond the scope of this paper. Dirty Paper Coding (DPC) [8] with a Gaussian coding ensemble achieves the capacity of the MIMO broadcast channel [36], but is difficult to implement in practice. The well-known linear Zero-Forcing Beamforming (ZFBF) achieves the same high-SNR capacity factor increase, with some fixed gap from optimal that can be reduced when the number of clients is large and the transmitter can dynamically select the clients to be served depending on their channel state information [19, 38]. A number of other precoding strategies, e.g., lattice reduction, regularized vector perturbation and generalized Tomlinson Harashima precoding, have been studied and the interested reader is referred to [29] and references therein. For the purposes of this paper ZFBF will be the primary method of interest because of its conceptual simplicity and good complexity/performance tradeoff.

Generalizing the idea of the MIMO broadcast channel, a vast literature has investigated distributed multiuser MIMO systems where several access points are connected to a common central processor through some backbone wired network and coordinate their signals to jointly serve a number of clients. If the backbone network has a sufficiently high bandwidth, the problem is conceptually identical to that of a single (distributed) multiple antenna terminal and therefore the same techniques of the MIMO broadcast channel can be applied. However, distributed multiuser MIMO presents several additional and non-trivial practical implementation problems related to the synchronization (phase and timing stability) of the separate coordinated access points, that need to maintain a very tight synchronization in order to be able to coherently precode (e.g., beamform) the signals to the clients without creating unacceptable multiuser self-interference.

A number of recent system implementations have made forays into the topics of multiuser MIMO transmission and distributed, frame aligned OFDM transmission. The benefits of using ZFBF as a precoding scheme have been examined in [2], in a system which consists of a single access point with multiple antennas hosted on the same radio board. The use of interference alignment and cancellation as a precoding technique, which does not require frame alignment or phase synchronization, has been illustrated in [15]. While this solution achieves a part of the potential spatial multiplexing gain, in order to realize the full spatial multiplexing with standard precoding techniques it is required to have tight phase synchronization [20, 34]

Simultaneous OFDM signal transmissions which are not separated in the spatial domain require precise frame alignment to maintain their frequency orthogonality. Two signals whose frame boundaries misalign by more than a cyclic prefix length cannot be reliably decoded due to interference leakage over the frequency domain during the decoding process. Frame alignment was used in SourceSync [25] in conjunction with space-time block coding in order to provide a diversity gain in a distributed MIMO downlink system. In Fine-Grained Channel Access [30], a similar technique allows for multiple independent clients to share the frequency band in fine increments, without a need for guard bands, resulting in a flexible OFDMA (OFDM with orthogonal multiple access) uplink implementation.

3. SYNCHRONIZATION IN DISTRIBUTED MIMO SYSTEMS

OFDM and Zero-Forcing Beamforming. OFDM has become the preferred digital signaling format in most modern broadband wireless networks, including WLANs IEEE 802.11a/g/n and 4G cellular systems. Its main characteristic is that it decomposes a frequency selective channel into a set of N parallel narrowband frequency-flat channels, where the number of frequency subcarriers N is a system design parameter. In a multiuser environment it has also a significant side advantage: as long as the different users' signals align in time with an offset smaller than a guard time interval called the cyclic prefix (CP), their symbols after OFDM demodulation will remain perfectly aligned on the time-frequency grid. In other words, the timing misalignment problem between user signals, which in single-carrier systems creates significant complications for joint processing of overlapping signals (e.g., multiuser detection [35], successive interference cancellation [32], Zig-Zag decoding [14]), completely disappears in the case of OFDM, provided that all users achieve a rather coarse timing alignment within the CP. [1]

In a point-to-point MIMO link with N_r receive antennas and N_t transmit antennas, the time-domain channel is represented by an $N_r \times N_t$ matrix of channel impulse responses. Thanks to OFDM, we can think of the channel in the frequency domain, such that the channel transfer function is described by a set of channel matrices of dimension $N_r \times N_t$, one for each of the N OFDM subcarriers. Because signals add linearly over the shared medium, the signal received at each client antenna is a linear combination of the signals sent from the access point's antennas. The receiver, having knowledge of the channel coefficients, is tasked with solving a linear system of N_r equations with N_t unknowns from which it can generally recover up to $\min(N_r, N_t)$ transmitted symbols (this is multiplexing gain, or degrees of freedom). Since the OFDM modulation breaks the spectrum into narrow subcarriers, this process is repeated on each independent subcarrier.

In contrast to point-to-point MIMO, in multiuser MIMO the receiver antennas are spatially separated and receivers are not generally able to communicate with one another. While before the receiver could find the sent symbols by just solving a set of linear equations, now each receiver has only one equation with several unknowns. In order to be able to solve for the variable of interest (the symbol intended for that receiver), we arrange that the contributions of all other unknowns cancel each other out in its particular equation. One of the techniques to achieve this is linear Zero-Forcing Beamforming (ZFBF).

In ZFBF, the transmitter multiplies the outgoing symbols by beamforming vectors such that the receivers see only their intended signals. For instance, let the received signal on a given subcarrier at user k be given by

$$y_k = h_{k,1}x_1 + h_{k,2}x_2 + \cdots + h_{k,N_t}x_{N_t} + z_k \qquad (1)$$

where $h_{k,j}$ is the channel coefficient from transmit antenna j to user k and z_k is additive white Gaussian noise. Then, the vector of all received signals can be written in matrix form as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \qquad (2)$$

where $\mathbf{H}$ has dimension $K \times N_t$, K denoting the number of single-antenna clients. Assuming $K \leq N_t$, we wish to find a matrix $\mathbf{V}$ such that $\mathbf{HV}$ is zero for all elements except the main diagonal, that is $\mathbf{HV} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$. When this occurs, then

$$\mathbf{y} = \mathbf{HVx} + \mathbf{z} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)\mathbf{x} + \mathbf{z}, \qquad (3)$$

assuring that each receiver k will see $y_k = \lambda_k x_k + z_k$, which is an independent channel with no interference.

When $\mathbf{H}$ has rank K (which is true with probability 1 for sufficiently rich propagation scattering environments typical of WLANs and for $K \leq N_t$) a column-normalized version of the Moore-Penrose Pseudoinverse generally yields the ZFBF matrix. This takes on the form

$$\mathbf{V} = \mathbf{H}^{\mathsf{H}}(\mathbf{H}\mathbf{H}^{\mathsf{H}})^{-1}\mathbf{\Lambda},$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$ ensures that the norm of each column of $\mathbf{V}$ is equal to 1, thus setting the total transmit power equal to $\mathrm{tr}(\mathrm{Cov}(\mathbf{Vx})) = \mathbb{E}[\|\mathbf{x}\|^2]$, i.e., equal to the power of the transmitted data vector $\mathbf{x}$. Since, by construction, $\mathbf{HV} = \mathbf{\Lambda}$ is a diagonal matrix, it follows that left multiplying the vector of user symbols with the beamforming matrix cancels out the symbol interference at the receivers.

Why Synchronization Is Needed. Time and phase synchronization are needed between transmitters in order for such precoding to work. Clearly, time synchronization is needed to coordinate transmissions, but because OFDM gives some leeway due to the cyclic prefix, this is a relatively coarse synchronization. Phase synchronization, however, is required since ZFBF relies on being able to precisely tune the phase of a signal arriving at a receiver. While a classic MIMO transmitter has all of its RF chains running on a single clock source, each access point in a distributed MIMO system has its own clock and thus the signal it produces drifts in phase with respect to the signals of the other access points. We will show that sufficiently accurate phase synchronization is necessary to make distributed multiuser MIMO a reality.

Why is distributed multiuser MIMO challenging? For simplicity of exposition, let us now consider a distributed multiuser MIMO scenario with two clients and two access points, each one with a single antenna. All of our considerations will apply equally to a more general scenario. For nomadic users, typical of WLAN scenarios, the channel changes quite slowly with time, so that we may assume that the channel impulse response is locally invariant with respect to time. In order to use ZFBF, we must estimate the channel matrix coefficients at each subcarrier for each transmitter/receiver antenna combination. A number of methods for estimating channel coefficients have been proposed, including feedback schemes (see [3] and the references therein) and exploiting uplink/downlink reciprocity in Time-Division Duplex (TDD) systems [18]. For simplicity of exposition, we will assume here that the channel

[1] Notice that typical CP length is between 16 to 64 times longer than the duration of an equivalent single-carrier symbol. For example, for a 20 MHz signal, as in standard 802.11g, the time-domain symbol interval is 50 ns, so that a typical CP length ranges between 0.7 and 3.2 μs.

estimates correspond perfectly to the real channel.

After the channels have been estimated, all of the access points send their channel estimates to a central server, which computes the precoding matrix. For each subcarrier $n = 1, \dots, N$, let

$$\mathbf{H}(n) = \begin{bmatrix} H_{11}(n) & H_{12}(n) \\ H_{21}(n) & H_{22}(n) \end{bmatrix} \tag{4}$$

denote the 2×2 downlink channel matrix between the two clients and the two access point antennas.

Let the precoding matrix $\mathbf{V}(n)$ of subcarrier n be such that $\mathbf{H}(n)\mathbf{V}(n) = \mathbf{\Lambda}(n) = \mathrm{diag}(\lambda_1(n), \lambda_2(n))$, as explained before. If timing and carrier phase reference remain unchanged from when the channel was estimated until the signal is transmitted, the received signal at the clients, on each subcarrier, can be written as

$$\mathbf{y}(n) = \mathbf{H}(n)\mathbf{V}(n)\mathbf{x}(n) + \mathbf{z}(n) = \mathbf{\Lambda}(n)\mathbf{x}(n) + \mathbf{z}(n) \tag{5}$$

Since the overall channel matrix $\mathbf{\Lambda}(n)$ is diagonal, we have achieved complete user separation, so that the access point can serve the two clients on the same downlink slot without interference. The spatial multiplexing gain in this case is 2, as two users are being served simultaneously on the same time-frequency resource.

Suppose now that the timing reference and carrier phase reference between the estimation and transmission slots of the two access point is not ideal. With perfect timing, the downlink channel from access point i to client j would have impulse response $h_{ij}(\tau)$. Instead, due to misalignment, the impulse response is $h_{ij}(\tau - \tau_i - \delta_j)e^{j(\phi_i + \theta_j)}$ where τ_i, ϕ_i are the timing and carrier phase shifts at access point i and δ_j, θ_j are the timing and carrier phase shifts at client j. For simplicity, assume that the timing shifts are integer multiples of the time-domain symbol interval T_s (otherwise the derivation is more complicated, involving the folded spectrum of the channel frequency response, but the end result is analogous). From the well-known rules of linearity and time-shift of the discrete Fourier transform, we arrive at the following expression for the effective channel matrix:

$$\widetilde{\mathbf{H}}(n) = \underbrace{\begin{bmatrix} e^{j(\frac{2\pi}{NT_s}\delta_1 n + \theta_1)} & 0 \\ 0 & e^{j(\frac{2\pi}{NT_s}\delta_2 n + \theta_2)} \end{bmatrix}}_{\mathbf{\Theta}(n)} \begin{bmatrix} H_{11}(n) & H_{12}(n) \\ H_{21}(n) & H_{22}(n) \end{bmatrix} \underbrace{\begin{bmatrix} e^{j(\frac{2\pi}{NT_s}\tau_1 n + \phi_1)} & 0 \\ 0 & e^{j(\frac{2\pi}{NT_s}\tau_2 n + \phi_2)} \end{bmatrix}}_{\mathbf{\Phi}(n)} \tag{6}$$

We notice that the diagonal matrix of phasors $\mathbf{\Theta}(n)$ multiplying the nominal channel matrix from the left poses no problems, since these phase shifts can be recovered individually by each client as in standard coherent communication [24]. In contrast, the diagonal matrix $\mathbf{\Phi}(n)$ multiplying from the right poses a significant problem, since in each receiver's equation each unknown will be further multiplied by a different random factor. In fact, since the server computes the MIMO precoding matrix $\mathbf{V}(n)$ based on $\mathbf{H}(n)$, it follows that when applied to the effective channel $\widetilde{\mathbf{H}}(n)$ in (6) the matrix multiplication $\widetilde{\mathbf{H}}(n)\mathbf{V}(n)$ is no longer necessarily diagonal. We conclude that the presence of timing and carrier phase misalignment between the estimation and transmission slots, at each individual access point, yields residual multiuser interference which may completely destroy the performance of a distributed multiuser MIMO system. To stress the importance of this aspect, we would like to make clear that the resulting signal mixing takes place over the actual transmission channel, making it impossible for the receivers to eliminate it.

Why Synchronization Is Possible. Any discussion on phase synchronization of distributed wireless transmitters must necessarily start with the mechanisms through which phase errors occur. Digital wireless transmission systems are constructed using a number of clock sources, among which the two most important ones are the sampling clock and the carrier clock. In a typical system, signals are created in a digital form in baseband at a sampling rate on the order of megahertz, then passed through a digital-to-analog converter (DAC). Through the use of interpolators and filters, the DAC creates a smooth analog waveform signal which is then multiplied by a sinusoidal carrier produced by the carrier clock. The result is a passband signal which is then sent over the antenna.

Wireless receivers, in turn, use a chain of signal multiplications and filters to create a baseband version of the passband signal received over the antenna. Some designs, such as the common superheterodyne receivers, use multiple high frequency clocks and convert a signal first to an intermediate frequency before bringing it back to baseband. Other designs simply use a carrier clock operating at the same nominal frequency as the carrier clock of the transmitter and perform the passage from passband to baseband in a single step. We will be focusing on such designs in the ensuing discussion. After baseband conversion, the signal is sampled and the resulting digital waveform is decoded.

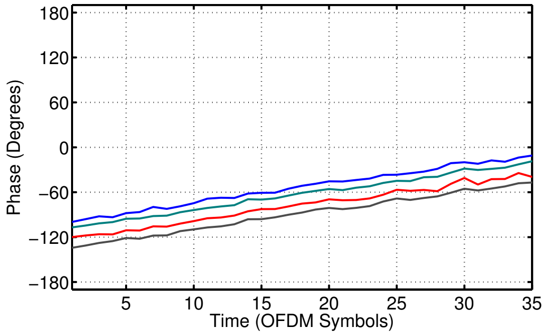

Figure 2: Pilot phases

There are four clocks in the signal path: the transmitter's sampling clock and carrier clock and the receiver's carrier clock and sampling clock. All four clocks manifest phase drift and jitter. The drift effect, when linear in time and happening at a relatively stable rate, can be assimilated to the presence of a carrier frequency offset.

We have considered the effects of a linear phase drift on an OFDM encoded packet in Equation 6. Denote by $\omega_n = \frac{2\pi n}{NT_s}$ the subcarrier frequency and with ω_c the carrier frequency. Let the timing error of the sampling clock be Δt_s and the timing error of the carrier clock be Δt_c and assume that they are on the same order of magnitude. The phase error due to the sampling clock will be $\omega_n \Delta t_s$ while the phase error due to the carrier clock will be $\phi_i = \omega_c \Delta t_c$. The term $e^{j(\frac{2\pi}{NT_s}\tau_i n+\phi_i)}$ in (6) can be rewritten as $e^{j(\omega_n \Delta t_s + \omega_c \Delta t_c)}$. Since ω_c is much greater than ω_n, the dominant phase rotation is due to the carrier clock and does not depend on the subcarrier frequency. Moreover, since time errors are additive, if the time error is approximately linear in time (linear clock drift) then the phase error will also be linear in time and almost equal for all subcarriers.

The assumptions behind the above statement are verified by the results presented in Figure 2. We have constructed an experiment in which a transmitter sends several tone signals, i.e., simple unmodulated sine waves, corresponding to several different subcarrier frequencies. In the absence of phase drift these tone signals would exhibit a constant phase when measured over several OFDM frames. In reality, the phase is not constant and the frame to frame phase drift of the tone signals can be measured and recorded. In the figure the phase drift has been plotted over the duration of a few tens of frames, a time length comparable to that of a packet transmission in a WLAN standard. As evidenced by these plots, our experiment confirms what was anticipated above: the drift is indeed linear and does not depend on the subcarrier frequency. This allows us to design a scheme for which the drift can be tracked and predicted.

The fact that the common phase drift of all subcarriers can be predicted by observing only a few pilots tones prompts the following approach to achieving phase synchronization between access points: a main access point (master) is chosen to transmit a reference signal consisting of several pilot tones placed outside the data transmission band, in a reserved portion of the system bandwidth. An initial channel probing header, transmitted by the master access point, is used by the other transmitters in order to get an initial phase estimate for each carrier. After this initial estimate is obtained, the phase estimates will be updated using the phase drift measured by tracking the pilot signals. After the initial channel estimation header, all access points start transmitting simultaneously in the data band, making use of the continuously updated phase estimates in order to create phase synchronous signals.

The achievable precision of this synchronization method depends on two main parameters: the SNR quality of the channel linking the secondary access points to the master access point and the jitter characteristics of the oscillator clocks. The impact of jitter can be estimated using the following back-of-the envelope calculation. Assume the use of an oscillator having a typical precision of 0.1 ppm (parts per million) over short time durations. The phase error of the synchronization circuit due to the oscillator can be estimated by multiplying the precision value with the time length of the synchronization loop. In our system, this loop has a time length corresponding to five OFDM symbols, or 80 microseconds. When assuming a carrier frequency of 2.4 GHz the resulting predicted phase offset is 3.5 degrees, which is more than adequate for our purposes as is evident from the experimental results that we present in Section 5. Capacity region calculations show that with this precision of synchronization, ZFBF can create, for an uniform user power allocation, parallel channels with up to a 27 dB SINR value.

4. IMPLEMENTING AIRSYNC

Software Radio Implementation. We have implemented AirSync as a digital circuit in the FPGA of the WARP radio platform [26]. The WARP radio is a modular software radio platform composed of a central motherboard hosting an FPGA and several daughterboards containing radio frequency (RF) front-ends. The entire timing of the platform is derived from only two reference oscillators, hosted on a separate clock board: a 20 MHz oscillator serving as a source for all sampling signals and a 40 MHz oscillator which feeds the carrier clock inputs of the transceivers present on the RF front-ends. The shared clocks assure that all signals sent and received using the different front-ends are phase synchronous. Phase synchronicity for all sent signals or for all received signals is a common characteristic of MIMO systems. However, the fact that the design of the

WARP ensures phase synchronicity among the sent and received signals, as opposed to using separate oscillators for modulation and demodulation, greatly simplifies the synchronization task. The system's data bandwidth is 5 MHz. We place the synchronization tones outside the data bandwidth, at about 7.5 MHz above and below the carrier frequency. The placement of the carriers allows us to exploit the adjustable baseband sender filter present in the transmit signal path in order to avoid, in the case of the pilot tones, any self-interference at the secondary transmitters.

We have implemented a complete system-on-chip design in the FPGA, taking advantage of the presence of hard-coded ASIC cores such as a PowerPC processor, a memory controller capable of supporting transfers through direct memory access over wide data buses and a gigabit Ethernet controller. Atop this system-on-chip architecture we have ported the NetBSD operating system and created drivers for all the hardware components hosted on the platform, capable of setting all system and radio board configuration parameters. The operating system runs locally but mounts a remote root filesystem through NFS. In the same system-on-chip architecture we integrated a signal processing component created in Simulink which provides interfaces for fast direct memory access. This later component is responsible for all the waveform processing and for the synthesis of a phase synchronous signal and interfaces directly with the digital ports of the radio front-ends. We interfaced the Ethernet controller and the signal processing component using an operating system kernel extension responsible for performing zero-copy, direct memory access data transfers between the two, with the purpose of passing back and forth waveform data at high rates between a host machine and the WARP platform. The large data rates needed (320 Mbps for a 10MHz wireless signal sampled with 16 bit precision) required optimizing the packet transfers into and out of the WARP. For example, consider the direct memory access ring associated with the receive end of the Ethernet controller on the board, which is shared between packets destined to the signal processing component and packets destined to the upper layers of the operating system stack. We do not release and reallocate the memory buffers occupied by packets destined to the signal processing component. Instead, we use a lazy garbage collection algorithm in order to reclaim these buffers when they are consumed in a timely manner or reallocate them at a later point if they are not consumed before the memory ring runs low on available memory buffers. The rationale for this particular optimization is that the overhead of managing the virtual-memory based reallocation of memory buffers of tens of thousands of packets every second would bring the processor of the software radio platform to a halt.

All transmitting WARP radios are connected to a central processing server through individual Ethernet connections operating at gigabit speeds. Most of the signal synthesis for the packet transmission is done offline, using Matlab code. We produce precoded packets in the form of frequency domain soft symbols. However, the synchronization step and the subsequent signal generation is left to the FPGA. The server, a fast machine with 32 processor cores and 64GB of RAM, encodes the transmitted packets and streams the resulting waveforms to the radios.

The Synchronization Circuit in Detail. AirSync operates similarly to other OFDM-based, distributed transmission systems such as SourceSync [25] or Fine Grained Channel Access [30], but extends them by achieving phase synchronization among transmitters. An important component of those systems, essential in order to avoid leakage from one carrier to another during the decoding process, is the realization of frame alignment that arranges frame starting points at the receivers within an interval shorter than a cyclic prefix length. In other words, the overlap of the frames sent by different senders must be greater than the length of a frame without cyclic prefix in order to allow the receiver to perform a full-length discrete Fourier transform on the received signal. Note that the use of zero-forcing does not relax this requirement. Zero-forcing is achieved by arranging the phases of several transmitted signals to sum up to almost zero at one of the receivers. The natural way in which these signals may add up to zero after applying the discrete Fourier transform is for the phase alignment between the different signals to be consistent for the whole duration of the transform. Thus the frames must overlap over the entire time interval associated with the transform. AirSync achieves frame synchronization through a technique used in block boundary detection, namely the insertion of pseudo-noise (PN) sequences in the master access point's packet header in order to allow the secondary transmitters and the receivers to obtain a time reference. For reasons that will become clear, achieving frame synchronization within the length of the cyclic prefix is a sufficient starting point for also achieving phase synchronization.

In the following we will say that two signals are phase synchronous when the pure tones (that is tones that have not been multiplied with a constellation symbol for data transmission) transmitted by their senders over each subcarrier, have a constant phase difference over the duration of successive frames and this difference can be known a priori by the senders. Naturally, due to the phase offsets induced by propagation delays the value of the phase difference depends on the location where the two signals are received. Thus the phase difference can be considered constant only when the two signals are compared at the same receive location.

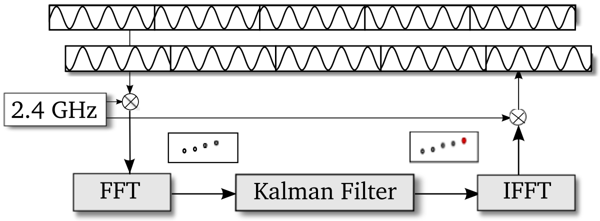

Figure 3: AirSync Schematic. The baseband signals are processed through an FFT which feeds phase estimates into a Kalman Filter. The IFFT produces a phase-adjusted data signal, with the same phase drift as the main transmitter. The modulation and demodulation use the same carrier clock.

AirSync implements the idea of observing phase drift using pilot tone signals. In order to reduce self-interference at the secondary transmitters, the tone signals are placed outside the data band, from which they are separated by a large guard interval. The secondary transmitters place an analog baseband filter around their data band further limiting their interference with the pilots. Self-interference could have been avoided using a number of other techniques such as antenna placement [7], digital compensation [10], or simply relying on the OFDMA-like property of a frame aligned system [30] and preventing the secondary transmitters from using the pilot subcarriers.

Figure 3 illustrates the process of creating a phase synchronous signal at the secondary transmitter while Figure 6 in Section 5.1 presents the initial synchronization sequence. The secondary transmitter overhears a packet sent by the primary transmitter and uses the initial PN sequence in order to determine the block boundary timing of this packet. Using a discrete Fourier transform the secondary transmitter decodes the successive frames of the incoming packet. It then employs the CORDIC algorithm on the complex-valued received soft symbols in order to obtain their phases in radians. The phases of the out-of-band pilot signals are tracked throughout the entire packet transmission in order to estimate the phase drift from the primary sender. The measurements from the four different pilots are averaged and passed through a simplified Kalman filter which maintains an accurate estimate and predicts, based on the current estimate, the phase drift after the passage of a few further frames. In addition, the header sent by the primary sender contains a number of channel estimation symbols, used to obtain an initial phase offset estimate for each subcarrier. As mentioned previously, the phase drift is almost identical for all carriers, therefore these two measurements suffice in order to predict the phase rotation induced by the main transmitter on any subcarrier tone for the entire period of a packet.

The phase estimates are used in synthesizing a synchronized signal. The secondary transmitter uses an inverse discrete Fourier transform, whose output frames are timed such that they align with the frames of the main sender's signal. For every subcarrier the secondary transmitter rotates the soft symbol to be sent by an angle corresponding to the subcarrier's estimated phase offset. The result is a tone that, while not having the same phase as the corresponding tone from the main transmitter, follows that tone at a fixed, pre-known phase difference.

The synchronization circuit could have been constructed in different ways. For example consider SourceSync [25], a recent work which has implemented frame alignment. AirSync differs in the implementation approach in three important points. SourceSync performs fine frequency offset correction. AirSync avoids this correction. Frequency offset correction prevents power leakage from neighboring carriers during decoding. Our synchronization circuit does not decode the subcarriers in the data band but only the pilot tones, for which power leakage from neighboring carriers is not a concern, and subsumes frequency correction on the transmit side by phase synchronization. Another design decision different from SourceSync is the use of a PN sequence for block boundary detection [33] instead of measuring the slope of the phase rotation induced by the timing misalignment between the sources on the decoded frames. The final difference from SourceSync is that since the senders are phase synchronous, the receivers do not need to monitor the evolution of the sender's pilots separately through joint channel estimation.

Centralized joint encoding. By transmitting phase synchronous signals from multiple access points we have created the equivalent of a distributed MIMO transmitter, capable of employing multiuser MIMO precoding strategies in order to transmit to multiple users at the same time. However, the use of multiple access points complicates the design of the transmitter system. For most of the precoding schemes available, the encoding of the waveforms to be transmitted over the antennas must be done jointly, since reaching a single user usually involves transmitting over multiple antennas. While in theory the joint encoding process could be duplicated at each access point given the binary information destined to each user, we chose to do the encoding only once, at a central server and send the resulting waveforms to each access point for transmission[2].

Our central server has an individual gigabit Ethernet connection to each of the WARP radios serving as access points. We divide the downlink time into slots and in each slot schedule for transmission a number of

[2] This approach is practical in enterprise networks where a number of access points are already connected to a common server.

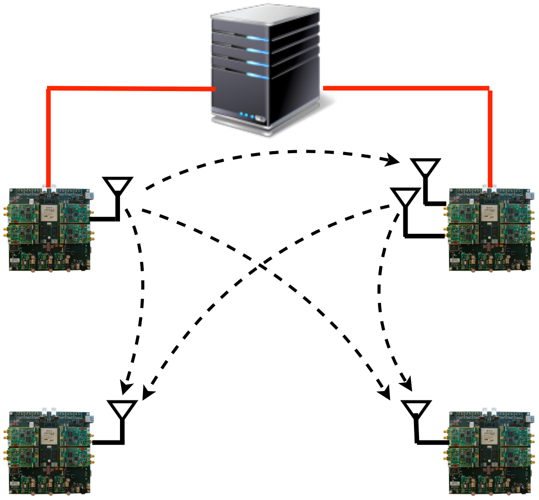

Figure 4: Testbed diagram. The central server is connected to the two transmitters, the main transmitter on the left and the secondary transmitter on the right.

packets destined to various users, according to an algorithm that will be presented in Section 6. The medium access encoding of the packets is presented in the same section. For each of the access points, the server computes the waveform of the signal to be transmitted in the next downlink slot. However, it does not perform any phase correction at this point. The only information used in the precoding is the data to be transmitted and the channel state information between each access point antenna and each user antenna. The server assumes that all access points are phase synchronous, like in a normal MIMO system. The server transmits their corresponding waveforms to all secondary transmitters and finishes by sending the last waveform to the primary transmitter. The primary transmitter starts transmitting right away and the secondary transmitters follow.

The design of AirSync ensures its scalability. There is no added overhead for synchronizing a larger number of secondary transmitters, while the overhead for channel estimation is the same as in regular MIMO systems.

In comparison to simple point-to-point transmission, AirSync uses about 10 more frames per packet in order to achieve synchronization. This number should be taken with a grain of salt in computing the overhead, since multiuser transmissions involve multiple packets broadcasted at the same time. When compared to a single packet duration, the overhead is about 4%.

5. PERFORMANCE EVALUATION

Our system setup is presented in Figure 4. It consists of a primary transmitter, a secondary transmitter and two receivers. The main sender uses a single RF front-end configured in transmit mode, placing an 18 MHz shaping filter around the transmitted signal. The secondary sender uses an RF front-end in receive mode and a second RF front-end in transmit mode, with a 12 MHz shaping filter. As mentioned previously, the pilots used in phase tracking are outside the secondary's transmission band, therefore the secondary transmitter will not interfere with the pilot signals from the main transmitter. The series of experiments is intended to test the accuracy of the synchronization and the efficiency of channel separation.

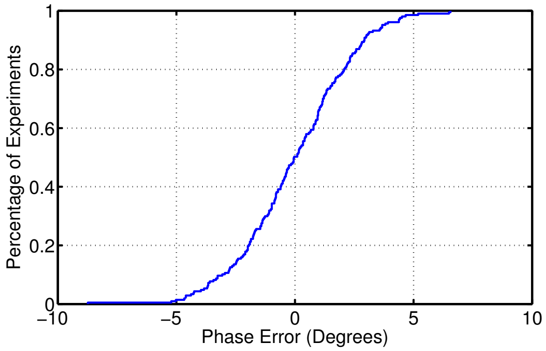

Figure 5: The Precision of the Phase Synchronization. AirSync achieves phase synchronization within a few degrees of the source signal.

5.1 Synchronization Accuracy

In this particular experiment we have placed the two transmitters and the two receivers at random locations. We placed a third RF front-end on the secondary sender and configured it in receive mode. The secondary transmitter samples its own synthesized signal over a wired feedback loop and compares it with the main transmitter's signal. The synchronization circuit measures and records the phase differences between these two signals. Since we use the primary transmission as a reference, in this experiment we do not broadcast the signal synthesized by the secondary transmitter in order to protect the primary transmission from unintended interference. We note that the use of a third RF front-end is not needed in the general case.

We have modified the synchronization circuit to produce a signal that is not only phase synchronous with that of the primary transmitter but has the exact same phase when observed from the secondary transmitter. To achieve this, the circuit estimates the phase rotation that is induced between the DAC of the secondary transmitter and the ADC through which the synthesized signal is resampled. It then compensates for this rotation by subtracting this value from the initial phase estimate. It is worth noting that this rotation corresponds to the propagation delay through the feedback circuit and is constant for different packet transmissions, as determined through measurements. The result was a synthesized signal that closely follows the phase

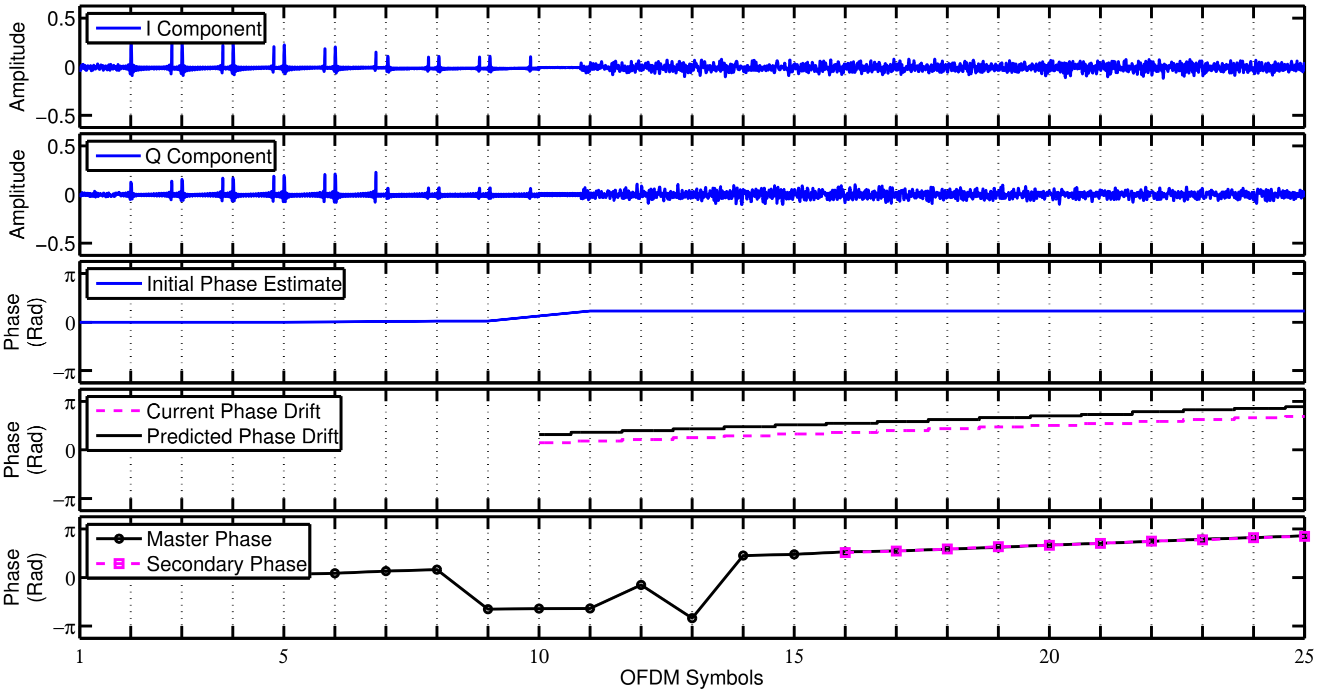

Figure 6: Phase Synchronization Acquisition. The secondary transmitter obtains an initial phase estimate. It then tracks the phase drift of the subcarriers and uses a Kalman filter to predict its value a few samples later.

of the signal broadcast by the master transmitter, as illustrated in Figure 6. The figure illustrates the initial phase acquisition process, the initial phase estimation, the tracking and estimation of the phase drift, as well as the synthesis of the new signal. The phase discontinuities appearing in the main transmitter's signal are due to the presence of PN sequence along with a temporary disturbance needed in order to tune the feedback circuit.

Figure 5 illustrates the CDF of the synchronization error between the secondary transmitter and the primary transmitter. The error is measured on a frame-to-frame basis using the feedback circuit. In decimal degree values, the standard deviation is 2.37 degrees. The 95th percentile of the synchronization error is at most 4.5 degrees.

We have measured the SNR value of the synchronization pilots in the signal received by the secondary transmitter to be around 28.5 dB. This is easily achievable between typically placed access points.

5.2 Beamforming gain

Our second experiment was done using the complete four radio setup with the secondary transmitter broadcasting a secondary signal over the air. We measured the channel coefficients between the two transmitters and the receivers using standard downlink channel estimation techniques and arranged the amplitudes and the phases of the transmitted signals such that at one of the receivers the amplitudes of the two transmitted signals would be equal while the phases would align. The maximal theoretic power gain over transmitting the two signals independently is 3.01dB. We compared the average power of the individual transmissions from the two senders to the average power of a beamformed joint transmission. Our measurements show an average gain of 2.98 dB, which is consistent with the precision of the synchronization determined in the previous experiment.

This result shows that for all practical purposes we are able to achieve the full beamforming gain in our testbed.

5.3 Zero-Forcing Accuracy

The following experiment measures the amount of power which is inadvertently leaked when using Zero-Forcing to non-targeted receivers due to synchronization errors. Again we have placed our radios at random locations in our testbed. We have estimated the channel coefficients and arranged for two equal amplitude tones from the two transmitters to sum as closely as possible to zero. The residual power is the leaked power due to angle mismatching. Figure 7 illustrates the CDF of this residual power for different measurements. The average power leaked is -24.46 dB of the total transmitted power.

This establishes that Zero-Forcing is capable of almost completely eliminating interference at non-targeted receiver locations.

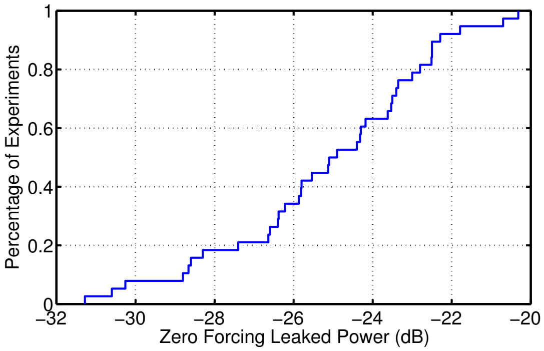

Figure 7: The Power Leakage of Zero-Forcing. The leaked power is significantly smaller than the total transmitted power, transforming each receiver's channel into a high SINR channel.

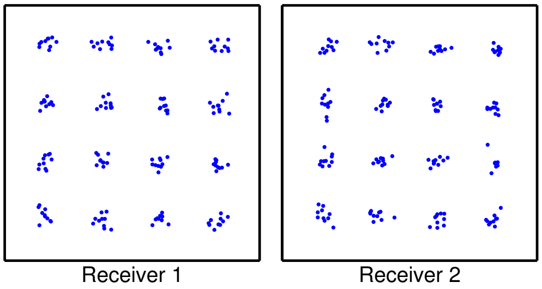

Figure 8: Scattering Diagram. The scattering diagram for two independent data streams transmitted concurrently demonstrates that AirSync achieves complete separation of the user channels.

5.4 Zero-Forcing Beamforming Data Transmission

The final experiment transmits data to the two receivers. We have used symbols chosen independently from a QAM-16 constellation at similar power levels. The scattering plots in Figure 8 illustrates the received signal at the two receivers.

The SINR values at the two receivers are 29 dB and 26 dB respectively. It is evident that the testbed achieves the full MIMO multiplexing gain.

6. MEDIUM ACCESS CONTROL

Given that we have achieved the necessary synchronization accuracy between access points, we turn to the large body of work on optimal scheduling for centralized multiuser MIMO systems (see for example [9, 19]). Inspired by this work, we propose a MAC layer that significantly departs from the classic networking layered architectural model and adopts a cross-layer “PHY/MAC” design strategy.

6.1 High level description

Time Division Duplexing. First, we consider the issue of allocating air time and frequency spectrum between the uplink and the downlink. We can choose between two natural strategies for separating the uplink from the downlink: time division duplex (TDD) and frequency division duplex (FDD). We choose to use TDD for two reasons. First, with TDD we can exploit channel reciprocity at the access point and measure the uplink channel, using pilots from the users, to infer the downlink channel as previously described. On the other hand, in FDD the uplink and downlink carrier frequencies are separated by much more than the channel coherence bandwidth, and therefore the channel matrix coefficients of the uplink and downlink channel are essentially statistically independent. Thus, in FDD no useful information about the downlink channel matrix can be learned from the uplink pilots. In this case, an explicit closed-loop channel estimation and feedback needs to be implemented, with a protocol overhead that increases linearly with the number of jointly precoded access point antennas [17]. Second, TDD is ideally suited for the transport of asymmetric traffic, as is typical in an enterprise WiFi scenario, whereas an FDD system provides less flexibility for managing different traffic patterns. We shall consider the scheduling of users in the uplink and downlink periods separately. In the uplink, clients compete for bandwidth using regular CSMA/CA. Thus, in the rest of this section we focus on the downlink.

Downlink scheduling. The central server keeps track of packet queue sizes and other readily available QoS information, e.g. the time since these queues have been served last. It then selects a subset of users to transmit to at each downlink time slot. At the start of the downlink period all access points send a jamming signal, causing the clients to backoff. The ensuing downlink transmission will silence the clients until the end of the downlink period. In the following we discuss in detail how the central server selects these users at each time slot.

The selection and power allocation problem for linear Zero-Forcing precoding has a rich literature (for example [9, 38]) which follows theoretical models similar to the one introduced in Section 3. Conceptually, this optimization problem can be solved by exhaustively searching over all feasible subsets of users, optimizing a weighted rate function under some general power constraints. In practice, greedy algorithms have proven to provide excellent results at moderate complexity [9, 19].

To simplify the design of the scheduling algorithm, we can use a greedy algorithm like the one described in [19]. However, real world considerations prompt a number of changes. Following the example of the de facto MIMO standard 802.11n, we may allocate the same power to all subcarriers instead of solving a complicated dual multiuser allocation and waterfilling problem. Note that this simplifies the optimization problem significantly. Second, to achieve fairness among flows the construction of the utility function must take into account the queue delays experienced by packets. This matters as it is well known that scheduling decisions which are solely based on queue sizes, while guaranteeing stability, lead to starvation and may lead to timeouts at higher layers, for example in TCP.

It is beyond the scope of this paper to further optimize the scheduling algorithm. In general, though, a non-flat spectrum will allow the system to get closer to the theoretically optimal performance. Instead, we turn our attention to the practical issues associated with our MAC protocol, and in particular, to the design of the packet headers.

6.2 Protocol Design

Our protocol design focuses on the downlink channel. Figure 9 presents a simplified schematic of the downlink data packets and corresponding uplink acknowledgments. The MAC layer packet design and the protocol's sequence of actions are tuned for enabling multiuser MIMO broadcasts. The crucial design constraint is to provide the central server with timely estimates of the channel state information for all clients to which it is about to transmit or which are considered for the next round of transmissions. For this purpose, we schedule downlink transmissions to closely follow uplink acknowledgments and require the clients to provide the server with channel estimates during the uplink period. The mechanism through which this is achieved will be described in the following paragraphs. The central server uses the uplink estimates to select a set of clients for the following transmission slots, according to the scheduling algorithms introduced earlier.

The downlink packet starts with a transmission from the main sender containing a pseudo-noise sequence used to achieve frame alignment by the transmitters and for block boundary detection by the receivers. The master access point then transmits the first set of channel estimation pilots which are used by the other access points to determine the initial phases of the subcarrier tones, as described in Section 4. After this point, all access points take part in the downlink transmission. The packet header that follows is broadcast to all clients, including the non-targeted ones, using the Alamouti encoding [1]. Due to phase alignment between transmitters, the clients do not need to track the secondary senders in order to decode this header. The MAC addresses of the hosts targeted in the current transmission and the MAC addresses of the clients that are required to provide the server with channel estimates during the next acknowledgment are the most important pieces of information contained in the header fields. The positions of the addresses in the header fields create an implicit ordering of the clients, which will be used in the uplink period. The following part of the header is an allocation map, similar to the one found in the LTE standard, which assigns carriers to small groups of different clients and specifies the constellations used in broadcasting to them. The header is followed by a second set of channel estimation pilots, transmitted this time around by all access points using ZFBF, which are used by all clients in order to obtain the channel estimates for their individual downlink channels. The clients use the downlink estimates together with the synchronization pilot tones in order to gain a lock on the subcarriers. The downlink transmission continues with payload transmission.

In current 802.11 MIMO implementations, the channel estimates are obtained using downlink pilots which are in turn quantized by the receivers and communicated back in numerical form to the transmitter. The quantization and communication steps incur a large overhead. Using the reciprocity property of wireless channels, we can reduce the complexity of the channel estimation process significantly. First, we prefer to perform uplink channel estimation since uplink estimates can be received simultaneously by all access points, reducing the number of pilot transmissions needed by a factor equal to the total number of access point antennas. Second, uplink estimates are sent using analog pilot signals in an unquantized form, leaving the quantization step to the access points. This reduces the overhead of the transmission significantly. Third, while the usual estimation pilots are full OFDM frames, we choose to send pulse-like signals, measure the channel response, and fill the non-significant taps with zeros before taking a Fourier transform in order to determine the frequency domain response. This insures that our pilots need to be spaced only by an interval that can accommodate a long channel response, i.e. the length of a cyclic prefix.

After the downlink transmission has finished, the clients who have been requested to send their channel estimates start sending these short estimation pilots in quick succession. We note that there is a large degree of similarity between the functioning of the downlink channel estimation for receive decode purposes and the uplink channel estimation step. The timing of the system remains unchanged during the uplink slot and the roles of the transmitters and the receivers are switched. The uplink pilots are followed by smart acknowledgments for the data packets sent using the technique detailed in [11].

We tested each component of the downlink and uplink protocol slots. However, since our radios do not switch from receive to transmit in a timely manner, we could not perform complete real-time MAC experiments.

Overhead. A note on the overhead of the above

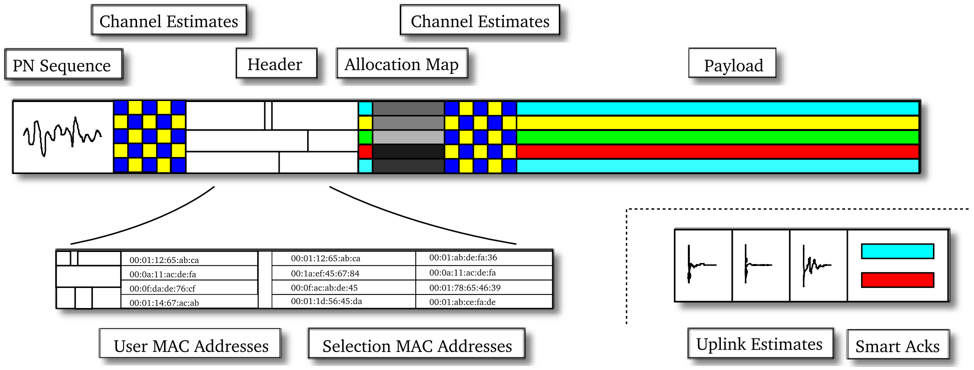

Figure 9: Packet Design. Downlink data packet (left) and uplink acknowledgment (bottom right).

MAC is in order. As those familiar with the PHY/MAC details of the 802.11 family of protocols would have recognized already, the overhead of our MAC is not more than that of 802.11n. The additional signaling overhead comes from requiring a few frames to predict the initial phase, and a few frames to dictate the MAC addresses of the nodes from which we wish to request channel state information for the next time slot. Even with very conservative estimates this will be less than a 20% increase in header time duration over that of a traditional 802.11 system. Note, however, that we get a bandwidth increase that grows almost linearly in the number of clients. This means that our overhead, normalized such that we consider the total control bits over the total data bits transmitted during a fixed airtime slot, is much less than in a traditional 802.11 system.

7. DISCUSSION AND FUTURE WORK

In the future, we plan to extend our experiments to a larger testbed. For demonstration purposes, we intend to showcase real time video streaming at high rates. In the rest of this section, we discuss approaches that go beyond the results discussed in the previous sections.

In this paper we used linear ZF precoding because of its conceptual and implementation simplicity, and near-optimal performance when allowing for flexible user selection [9,38]. In the future we plan to experiment with more sophisticated precoding techniques, such as regularized ZF [22], lattice reduction precoding [37] and modified Tomlinson-Harashima precoding [5], which can be regarded as a viable and very efficient approximation of the capacity-achieving Dirty Paper Coding (DPC) scheme [8]. The relative merit of these schemes is known from a theoretical viewpoint, but a thorough comparison on the basis of an actual SDR implementation is not available.

In wireless communication systems, different rates are usually supported using different codes. The current standard, 802.11n, offers many code combinations to fully utilize the capacity of the MIMO channel. Since a multiuser MIMO system serves multiple users in the same time slot, an even larger set of rates and codes would have to be supported for efficiently using capacity. In this case, an attractive and innovative approach would be the use of *rateless codes* (e.g., Raptor codes [12,28] and the recently proposed Spinal codes [23]) at the physical layer, in a so-called *Incremental Redundancy* (IR) configuration (see [6,21,27]), as already exemplified by Strider [16], to decrease the signaling and retransmission overhead. This is another interesting direction that we plan to pursue as future work.

8. ADDITIONAL AUTHORS

9. REFERENCES

[1] S. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE J. Sel. Areas Commun.*, 16(8):1451–1458, 1998.

[2] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *ACM MobiCom*, Chicago, IL, 2010.

[3] G. Caire, N. Jindal, M. Kobayashi, and N. Ravindran. Multiuser MIMO achievable rates with downlink training and channel state feedback. *IEEE Trans. Inf. Theory*, 56(6):2845–2866, 2010.

[4] G. Caire and S. Shamai. On the achievable throughput of a multiantenna gaussian broadcast channel. *IEEE Trans. Inf. Theory*, 49(7):1691 – 1706, Jul. 2003.

[5] G. Caire, S. S. Shamai, A. Shokrollahi, and S. Verdu. Fountain codes for lossless data compression. In *AMS DIMACS Workshop on Algebraic Coding Theory and Information Theory*, Jun. 2005.

[6] G. Caire and D. Tuninetti. The throughput of hybrid-arq protocols for the gaussian collision channel. *IEEE Trans. Inf. Theory*, 47(5):1971 –1988, Jul. 2001.

[7] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *IEEE MobiCom*, Chicago, IL, 2010.

[8] M. Costa. Writing on dirty paper (corresp.). *IEEE Trans. Inf. Theory*, 29(3):439–441, May 1983.

[9] G. Dimic and N. Sidiropoulos. On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm. *IEEE Trans. Signal Process.*, 53(10):3857 – 3868, Oct. 2005.

[10] M. Duarte, C. Dick, and A. Sabharwal. Experiment-driven characterization of full-duplex wireless systems. *CoRR*, abs/1107.1276, 2011.

[11] A. Dutta, D. Saha, D. Grunwald, and D. Sicker. SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks. In *ACM SIGCOMM*, Barcelona, Spain, 2009.

[12] O. Etesami and A. Shokrollahi. Raptor codes on binary memoryless symmetric channels. *IEEE Trans. Inf. Theory*, 52(5):2033 – 2051, May 2006.

[13] G. Foschini and M. Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6:311–335, 1998.

[14] S. Gollakota and D. Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *ACM SIGCOMM*, Seattle, WA, 2008.

[15] S. Gollakota, S. D. Perli, and D. Katabi. Interference alignment and cancellation. In *ACM SIGCOMM*, Barcelona, Spain, 2009.

[16] A. Gudipati and S. Katti. Strider: automatic rate adaptation and collision handling. In *ACM SIGCOMM*, Toronto, Ontario, Canada, 2011.

[17] J. Jose, A. Ashikhmin, P. Whiting, and S. Vishwanath. Channel estimation and linear precoding in multiuser multiple-antenna TDD systems. *IEEE Trans. Veh. Technol.*, 60(5):2102 –2116, Jun. 2011.

[18] J. Jose, A. Ashikhmint, P. Whiting, and S. Vishwanath. Scheduling and pre-conditioning in multi-user MIMO TDD systems. In *IEEE ICC*, Beijing, China, 2008.

[19] M. Kobayashi and G. Caire. Joint beamforming and scheduling for a MIMO downlink with random arrivals. In *IEEE ISIT*, Seattle, WA, Jul. 2006.

[20] A. Lapidoth, S. Shamai, and M. A. Wigger. On the capacity of fading MIMO broadcast channels with imperfect transmitter side-information. In *43rd Annual Allerton Conference*, Monticello, IL, 2005.

[21] C. Lott, O. Milenkovic, and E. Soljanin. Hybrid ARQ: Theory, state of the art and future directions. In *IEEE ITW on Information Theory for Wireless Networks*, Solstrand, Norway, Jul. 2007.

[22] C. Peel, B. Hochwald, and A. Swindlehurst. A vector-perturbation technique for near-capacity multiantenna multiuser communication-part i: channel inversion and regularization. *IEEE Trans. Commun.*, 53(1):195 – 202, Jan. 2005.

[23] J. Perry, H. Balakrishnan, and D. Shah. Rateless spinal codes. In *ACM HotNets*, Cambridge, Massachusetts, 2011.

[24] J. Proakis and M. Salehi. *Digital communications.* McGraw-Hill, New York, NY, 2007.

[25] H. Rahul, H. Hassanieh, and D. Katabi. SourceSync: a distributed wireless architecture for exploiting sender diversity. In *ACM SIGCOMM*, New Delhi, India, 2010.

[26] Rice University. Rice university warp project.

[27] S. Sesia, G. Caire, and G. Vivier. Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes. *IEEE Trans. Commun.*, 52(8):1311 – 1321, Aug. 2004.

[28] A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551 –2567, Jun. 2006.

[29] Q. Spencer, C. Peel, A. Swindlehurst, and M. Haardt. An introduction to the multi-user MIMO downlink. *IEEE Commun. Mag.*, 42(10):60–67, 2004.

[30] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang. Fine-grained channel access in wireless LAN. In *ACM SIGCOMM*, New Delhi, India, 2010.

[31] E. Telatar. Capacity of multi-antenna gaussian channels. *European Transactions on Telecommunications*, 10(6):585–595, 1999.

[32] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, New York, NY, 2005.

[33] F. Tufvesson, O. Edfors, and M. Faulkner. Time and frequency synchronization for OFDM using PN-sequence preambles. In *IEEE VTC*, 1999.

[34] C. S. Vaze and M. K. Varanasi. The degrees of freedom regions of MIMO broadcast, interference, and cognitive radio channels with no CSIT. *CoRR*, abs/0909.5424, 2009.

[35] S. Verdu. *Multiuser Detection.* Cambridge University Press, New York, NY, 1998.

[36] H. Weingarten, Y. Steinberg, and S. Shamai. The capacity region of the gaussian multiple-input multiple-output broadcast channel. *IEEE Trans. Inf. Theory*, 52(9):3936 –3964, Sept. 2006.

[37] H. Yao and G. Wornell. Lattice-reduction-aided detectors for MIMO communication systems. In *IEEE GLOBECOM*, Hsinchu, Taiwan, Nov. 2002.

[38] T. Yoo and A. Goldsmith. On the optimality of multiantenna broadcast scheduling using

zero-forcing beamforming. *IEEE J. Sel. Areas Commun.*, 24(3):528 – 541, Mar. 2006.