SAM: Enabling Practical Spatial Multiple Access in Wireless LAN

Kun Tan
Microsoft Research Asia
Beijing, China
kuntan@microsoft.com

He Liu*
Microsoft Research Asia
and Tsinghua University
Beijing, China
v-hliu@microsoft.com

Ji Fang*
Microsoft Research Asia
and Beijing Jiaotong University
Beijing, China
v-fangji@microsoft.com

Wei Wang, Jiansong Zhang
Microsoft Research Asia
Beijing, China
{wei.wang, jiazhang}
@microsoft.com

Mi Chen*
Microsoft Research Asia
and SouthEast University
Beijing, China

Geoffrey M. Voelker
Univ. of California San Diego
La Jolla, CA, USA
voelker@cs.ucsd.edu

ABSTRACT

Spatial multiple access holds the promise to boost the capacity of wireless networks when an access point has multiple antennas. Due to the asynchronous and uncontrolled nature of wireless LANs, conventional MIMO technology does not work efficiently when concurrent transmissions from multiple stations are uncoordinated. In this paper, we present the design and implementation of a cross-layer system, called SAM, that addresses the challenges of enabling spatial multiple access for multiple devices in a random access network like WLAN. SAM uses a *chain-decoding* technique to reliably recover the channel parameters for each device, and iteratively decode concurrent frames with misaligned symbol timings and frequency offsets. We propose a new MAC protocol, called CCMA, to enable concurrent transmissions by different mobile stations while remaining backward compatible with 802.11. Finally, we implement the PHY and MAC layer of SAM using the Sora high-performance software radio platform. Our evaluation results under real wireless conditions show that SAM can improve network uplink throughput by 70% with two antennas over 802.11.

Categories and Subject Descriptors

C.2.1 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Design, Experimentation, Performance

*This work is done when He Liu, Ji Fang and Mi Chen are research interns in Microsoft Research Asia.

MobiCom'09, September 20–25, 2009, Beijing, China.

Keywords

Spatial Division Multiplexing, MIMO, Software radio, wireless

1. INTRODUCTION

Multiple-input and multiple-output (MIMO) is an emerging technology to significantly boost network capacity by exploiting the spatial properties of wireless channels. It has been included in several wireless standards, notably IEEE 802.11n, for which many commercial devices already exist. For example, Figure 1(a) shows a device (station 3) with two antennas. When communicating with the AP with multiple antennas, station 3 can transmit separate frames on each antenna simultaneously, and thus potentially improve its link capacity by a factor of two.

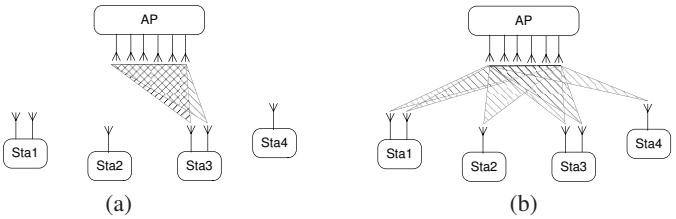


Figure 1: (a) MIMO and (b) "virtual MIMO" (spatial multiple access).

With single-user MIMO, the capacity improvement is bounded by the number of transmitter or receiver antennas, whichever is smaller. In practice, due to size, cost, and power limitations, mobile stations generally have only a few antennas (*e.g.*, most 802.11n devices have only two antennas). However, APs do not have the same constraints as mobile stations, can be generously provisioned with resources, and potentially have a much larger number of MIMO radios. Nevertheless, the network capacity will not improve beyond the link capacity as constrained by the number of antennas at the mobile station.

Previous results in information theory [17] indicate that it is possible for multiple stations to form a "virtual MIMO" system in which the stations transmit simultaneously (spatial multiple access). In this kind of network, the AP may still be able to decode all frames correctly as long as the number of concurrent frames is less than the number of antennas at the AP. As shown in Figure 1(b),

with spatial multi-access, all stations can transmit simultaneously to make full use of the AP's antennas. Thus, the network capacity can increase linearly with the number of antennas at the AP, precisely the device in the network that can best accommodate the cost, size, and power of a relatively large number of radios.

In this paper, we presents SAM, a practical system that enables spatial multiple access for uplink traffic in uncontrolled wireless LANs, and thereby indirectly increases the throughput of downlink traffic [1]. In SAM, there is no need for mobile stations to tightly synchronize their carrier frequency or timing to the APs, nor do devices need to exchange channel state information. With SAM, mobile stations can coordinate their transmissions in a fully distributed fashion, thereby utilizing the spatial wireless channel without relying upon AP coordination. Such distributed coordination is a good match to the requirements of typical wireless LANs, as well as adhoc networks where multiple mobile stations may communicate in a peer-to-peer mode.

Implementing spatial multiple access in wireless LAN confronts a series of technical challenges, ranging from the inability to physically align symbol timing or carrier frequency among different stations (necessary for APs to correctly decode concurrent frames), to the lack of a MAC protocol to encourage and coordinate concurrent accesses. SAM addresses these challenges with two innovative techniques, a new transmission and decoding structure called *chain-decoding* that exploits the asynchronous nature of simultaneous transmissions to fully decode all concurrent frames, and a novel distributed MAC protocol called Carrier Counting Multiple Access (CCMA) where stations independently observe and compete for concurrent transmission opportunities.

We have implemented SAM in a high-speed software radio platform. In our current implementation, SAM operates over a real IEEE 802.11b network with four different modulation rates: 1, 2, 5.5 and 11Mbps. With each station equipped with one antenna and an AP with two antennas, we show SAM improves the network uplink capacity by 45–76% at 5.5Mbps, and 31–61% at 11Mbps modulation rate. Further, SAM remains backward compatible and interoperable with standard 802.11 implementations. Indeed, in our experiments one of the mobile stations uses a commercial 802.11 NIC.

In summary, this paper makes three major contributions with SAM: (1) the design and implementation of the chain-decoding structure that can effectively decode concurrent transmissions from asynchronous senders in wireless LANs; (2) the design and implementation of the Carrier Counting Multiple Access (CCMA) MAC protocol to coordinate spatial multi-access; and (3) an experimental evaluation of SAM over a real IEEE 802.11b network. To the best of our knowledge, SAM is the first working system to enable spatial multiple access in a high-speed wide-band wireless LAN environment.

The rest of the paper is organized as follows. Section 2 provides background on MIMO and spatial multiple access, and lays out the technical challenges in implementing them in high-speed and wide-band wireless systems. We next describe in detail the two core components of SAM, namely chain-decoding and CCMA, in Sections 3 and 4, respectively. After describing our implementation of SAM using a high-performance SDR platform in Section 5, we evaluate the performance of SAM in Section 6. Finally, Section 7 discusses related work and Section 8 concludes.

[1]Admittedly somewhat irregular, we simply preferred the suggestive name SAM over the strict acronym SMA.

2. BACKGROUND AND CHALLENGES

In this section we provide background on wireless communication fundamentals and single-user multiple-input multiple-output (MIMO) systems. We then describe spatial multiple access and the challenges of implementing a practical spatial multi-access system in a typical wireless LAN environment.

2.1 A Wireless Communication Primer

In digital communication, a baseband wireless signal is represented as a series of discrete complex numbers. Each number, called a symbol, represents certain bits of information. For example, in BPSK, symbol $e^{j\pi}$ represents bit "0" and symbol e^{j0} represents bit "1". When transmitting, the series of wireless symbols is fed to a D/A convertor in a fixed interval T. We denote $x[n]$ the complex symbol transmitted at time nT. Then, the baseband signal is multiplied by a high-frequency carrier signal and emitted through an antenna. At the receiver, the baseband signal is first separated from the carrier and then digitized by an A/D convertor into discrete-time samples. We denote $y[n]$ the sampled value at the receiver. In general, we can also use vector $\boldsymbol{y}$ to present all received symbols, and vector $\boldsymbol{x}$ for all transmitted symbols. In a simple flat fading channel, *e.g.* narrow band wireless, we have

$$y[n] = h[n]x[n] + w[n], \tag{1}$$

where $h[n] = \alpha[n]e^{-j\theta[n]}$ is a single complex number representing the channel attenuation ($\alpha[n]$) and the phase shift ($\theta[n]$) on the transmitted symbol. In principle, a receiver must correctly recover the original transmitted symbols $\boldsymbol{x}$ from the distorted receive symbols $\boldsymbol{y}$.

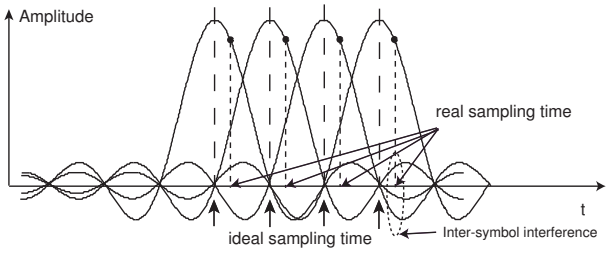


Figure 2: Symbol timing offset and inter-symbol interference.

Before the receiver can demodulate the received signals, it must have accurate knowledge of the wireless channel parameters. First, the receiver must know the exact symbol timing, *i.e.*, the signal's "ideal sampling points" which will provide the best reading for the symbols and minimize *intersymbol interference* (ISI), a phenomenon where the wave of a symbol spills over onto the neighboring symbols (Figure 2). Due to propagation and Doppler effects, and the receiver's sampling clock being independent of the sender's, it is highly unlikely that the receiver will sample at the exact symbol time. The receiver must either accurately track this timing offset, or it may apply *over-sampling* to take samples at a smaller interval (an integer fraction of T) and interpolate the ideal sample values as if they were taken at the ideal sampling points.

Second, in a real system the high frequency carrier signal is generated with an electronic circuit called a crystal oscillator. Under current engineering constraints, all oscillators have a small variation which will cause a slight difference between the sender and receiver's carrier frequency and a phase rotation in any received signals. This variation will accumulate over time and cause decoding failure. For example, typical oscillators used in wireless communications today stablize at the order of parts per million (ppm),

meaning that severe phase distortions can happen at a millisecond timescale in 2.4GHz. In a real system this *carrier frequency offset* must be precisely estimated and compensated for when solving Equation 1 above.

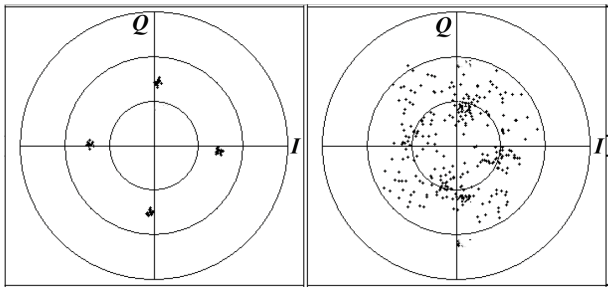


Figure 3: Illustration of multi-path fading. The left figure shows the transmitted signal, and the right figure shows the actual received signal in a real environment.

Third, wide-band communication is always subject to multi-path fading [15] (see Figure 3 for an illustration). That is, the wireless channel instead should be represented by a vector

$$\boldsymbol{h} = \{h_{-L}, ..., h_0, ..., h_L\},$$

where each component represents a path that the signal traverses. Then the received signal $y[n]$ becomes $y[n] = \boldsymbol{h} * x[n] + w[n]$ where $*$ is the convolution operation. To revert the distortion due to multi-path fading, a receiver must apply a linear filter $\boldsymbol{c}$ such that

$$\hat{x}[n] = \boldsymbol{c} * y[n]. \quad (2)$$

Such an *equalizer* must be first properly trained under the same multi-path fading channel.

To summarize, a receiver must first know these critical system parameters before it can reliably decode a frame. To facilitate this, many wireless systems such as WiFi adopt a self-training scheme: it prepends each transmission frame with a series of known training symbols called the *preamble*. If the preamble is received in a clean channel, the receiver can accurately estimate the symbol timing, carrier frequency offset, and channel coefficients (equalizer) necessary to properly decode the subsequent frame.

2.2 A MIMO Primer

In a MIMO system, if antennas are separated properly, each transmitting and receiving antenna pair can have a channel different from the others. For example, in a 2x2 MIMO system illustrated in Figure 4, two simultaneous transmissions can occur. Two receiving antennas can be represented with the following two linear equations:

$$\begin{aligned} y_1[n] &= \boldsymbol{h}_{11} * x_1[n] + \boldsymbol{h}_{12} * x_2[n] + w_1[n] \\ y_2[n] &= \boldsymbol{h}_{21} * x_1[n] + \boldsymbol{h}_{22} * x_2[n] + w_2[n] \end{aligned} \quad (3)$$

where $\boldsymbol{w_i}$ is the noise, and $\boldsymbol{h}_{ij}$is the complex channel coefficient vector for the multi-path channel between the sending and receiving antenna pair. As long as the channels are orthogonal, meaning that the equations will be linearly independent, both $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ can be resolved.

For the same reason as described in the previous subsection, a MIMO receiver must first acquire knowledge of all channels between any transmitting/receiving antenna pairs, including their symbol timing, frequency offset, and channel coefficients. This acquisition can be achieved with a simple extension of the preamble scheme, as in 802.11n where each transmitting antenna sends

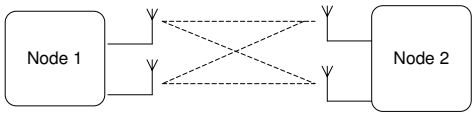


Figure 4: Example of single-user 2x2 MIMO.

a separate preamble, one-by-one and non-overlapping, so that each receiving antenna can receive a clean preamble from each transmitting antenna.

2.3 Spatial Multiple Access and Technical Challenges

The number of concurrent transmissions in a MIMO system is always limited to the number of antennas at the sender or the receiver, whichever is smaller. If an AP in a wireless LAN has more antennas, it makes sense to allow more stations to access the wireless channel concurrently to make full use of the number of receiving antennas at the AP. In the example of Figure 1(b), the AP has six antennas and each station has only one or two. In this case, the four stations can form a "virtual MIMO" system and transmit six frames simultaneously to the AP, achieving six times the capacity in theory.

To implement this "virtual MIMO" system in a wireless LAN, there are several difficult technical challenges. First, the transmitting antennas are now distributed across several independent sending stations, and there is no guarantee of an interference-free transmission of their preambles. This preamble interference will significantly affect the accuracy of channel estimations and may result in the loss of all concurrent frames. One naive solution is to apply excessively long preambles to each frame to average out the impact of interference, but doing so will impose significant overhead in the communication system. A more practical approach is to develop a distributed preamble scheme to enable interference-free decoding despite interference from other concurrent transmissions, while minimizing the overhead of the preambles on the concurrent channel.

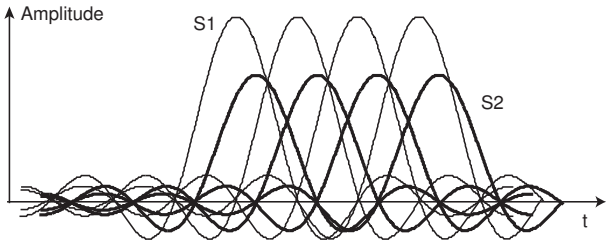


Figure 5: Unsolvable symbol timing mis-match in spatial multi-access.

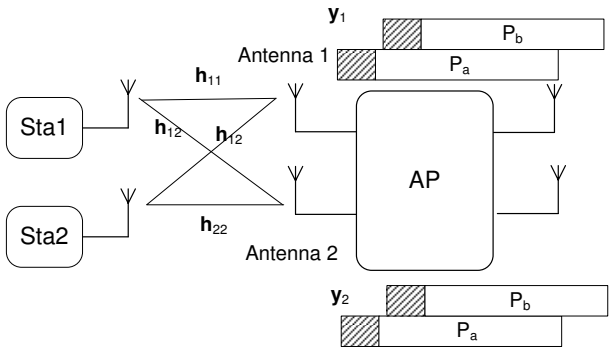
Second, the fact that transmitting antennas are not driven from the same oscillators introduces further challenges in decoding concurrent frames. In a simple MIMO system, all transmitting antennas have the same symbol timing and frequency offset. In a "virtual MIMO" system, however, when signals from multiple sending stations superimpose together at one of the AP's receiving antenna, neither their symbol timings nor their frequency offsets will align together. In particular, there may no longer exist "ideal sampling points". As illustrated in Figure 5, at the times when the intersymbol interferences from one signal are minimized, the intersymbol interferences from another signal become high. That is, no matter how much over-sampling the receiver can perform, the samples

are always distorted by interference. This misalignment essentially means Equation 3 can never be satisfied in spatial multi-access, and a conventional single user MIMO decoder will not work. Prior work on this problem focuses on tight synchronization among all sending stations [2], but such an approach requires very complicated electronics that may not suit environments like loosely controlled wireless LANs.

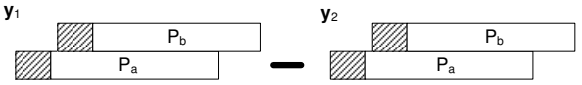
Both problems must be solved to enable spatial multi-access in wireless LANs. The fundamental challenge derives from the distributed nature of concurrent transmissions from unsynchronized stations. This challenge calls for a cross-layer PHY/MAC solution to exploit spatial multiplexing in a distributed and unsynchronized method.

3. CHAIN-DECODING

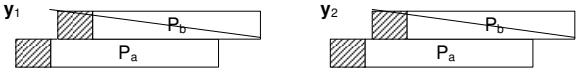
SAM enables spatial multiple access in wireless LANs by allowing multiple concurrent transmission to be mostly overlapped: a latter frame will start transmission after the end of the preamble of a former frame. In Figure 6(a), we show an example of two frames that are partially overlapped in this manner. SAM uses a decoding structure, called *chain-decoding*, that can reliably decode the frames in an overlapping transmission. In this section, we describe the steps involved in chain-decoding overlapped transmissions, and in the next section we describe a straightforward backwards-compatible MAC protocol that enables stations to take advantage of chain-decoding. For the sake of simplicity, in the following we focus on the case where only the AP has multiple antennas and all mobile stations have one antenna. It is straightforward to extended the scenario to the case where mobile nodes in the network also have multiple antennas.



(a) Two stations transmit two frames concurrently.



(b) AP performs *interference nullifying* to remove the signal of P_a. Then, P_b is ready for decoding.



(c) After decoding P_b, AP applies *interference cancelation* to remove the signal of P_b. Then, P_a is ready for decoding.

Figure 6: Illustration of chain decoding.

We start with the basic example in Figure 6(a) where two stations transmit concurrently. Without loss of generality, we assume that AP selects two received signals from antenna 1 and 2 (labeled $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$, respectively) for decoding. Note that the AP may have more than two antennas, and it can further improve reliability by using the redundant antennas for diversity selection or maximal radio combining (MRC) [3]. Later, in Section 3.3, we generalize chain-decoding to the case where more stations spatially access the channel.

When the AP receives these two overlapped transmissions at its two antennas, it performs the following two steps to decode each frame:

1. **Interference nullifying.** In the first step, the AP nullifies the signal of P_a from both received signals, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Then, it will obtain a version of the signal containing only P_b which is ready for decoding.

2. **Interference cancelation.** After P_b is successfully decoded, it can be re-encoded and canceled from the original signals, $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. Such cancelation will result in signals containing only P_a. Thus, P_a is then ready for decoding.

3.1 Interference Nullifying

From the AP's point of view, the received signals $\boldsymbol{y}_i$ are superpositions of the two signals of P_a and P_b, denoted as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, that pass different wireless channels. We can use a vector to present a received signal. In Figure 7(a), we show the relation among the received signal $\boldsymbol{y}_i$ and the channel distorted signal versions of $\boldsymbol{x}_i$. The idea of interference nullifying (IN) is to find two proper linear transforms for $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ such that, after transformation, the signals for one frame, say $\boldsymbol{x}_1$, are aligned with exactly the same direction and scale, as shown in Figure 7(b). Then, a substraction of these two transformed signals will completely remove $\boldsymbol{x}_1$. The result is another signal $\boldsymbol{y}_3$ which contains only the information of $\boldsymbol{x}_2$ and is ready for decoding.

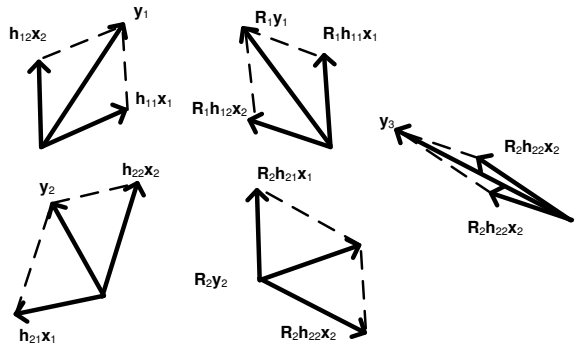


Figure 7: Illustration of interference nullifying.

Although there can be multiple ways to align the signals, the most natural way is to recover the signals to the original $\boldsymbol{x}_1$ that is transmitted, which is an operation that should be done in any communication system. Since we have a clean preamble of P_a, we can easily estimate the necessary system parameters of P_a, including symbol timing and carrier frequency offset, using standard mechanisms for single-user communication [11]. However, particular attention should be paid to the design of the equalizer that removes the distortion due to multi-path fading.

(a) Channel equalizer for interference nullifying

As mentioned earlier, an equalizer is a linear filter $\boldsymbol{c} = \{c_{-L}, ..., c_L\}$ that reverses the impact of a multi-path wireless channel. Mathematically, an optimal equalizer minimizes the difference between the recovered signals and the signals that were transmitted, as:

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}} ||\boldsymbol{x} - \boldsymbol{c} * \boldsymbol{y}||,$$

1: Initialize $\{c_i\}$:
2: $c_0 = 1$
3: $c_j = 0, j \neq 0$
4:
5: Training:
6: **for** each sample i **do**
7: $\quad \hat{x}_i = \sum_{l=-L}^{L} c_l y_{i-l}$
8: $\quad \epsilon = x_i - \hat{x}_i$
9: $\quad$ **for** $j = -L..L$ **do**
10: $\quad\quad c_j = c_j - \Delta \cdot \frac{\epsilon y^*_{i-j}}{|y_{i-j}|^2}$
11: $\quad$ **end for**
12: **end for**

(a)

1: Initializer $\{c_i\}$:
2: $c_0 = 1$
3: $c_j = 0, j \neq 0$
4:
5: Training:
6: **for** each sample i **do**
7: $\quad$ **for** $k = 0..K-1$ **do**
8: $\quad\quad \hat{x}_{i+k} = \sum_{l=-L}^{L} c_l y_{i+k-l}$
9: $\quad\quad \epsilon_k = x_{i+k} - \hat{x}_{i+k}$
10: $\quad\quad \delta_j = 0, j = -L..L$
11: $\quad\quad w = 0$
12: $\quad\quad$ **for** $j = -L..L$ **do**
13: $\quad\quad\quad \delta_j = \delta_j + \Delta \cdot \epsilon_k y^*_{i+k-j}$
14: $\quad\quad\quad w = w + |y_{i+k-j}|^2$
15: $\quad\quad$ **end for**
16: $\quad$ **end for**
17: $\quad$ **for** $j = -L..L$ **do**
18: $\quad\quad c_j = c_j - \delta_j / w$
19: $\quad$ **end for**
20: **end for**

(b)

Figure 8: Pseudo-code for equalizer training algorithms. The same algorithm also applies to channel estimation with the exchange of input and output.

where $\boldsymbol{x}$ is the transmitted signal, $\boldsymbol{y}$ is the received signal, $||.||$ represents the 2-norm, and "$*$" is the convolution operation.

To obtain each component value of $\boldsymbol{c}$, the receiver uses the known preamble to train the equalizer with an adaptive searching algorithm. Figure 8(a) outlines the Normalized Least Mean Square algorithm for an adaptive equalizer [15]. However, a conventional equalizer works at the symbol rate and it can only capture the signal path with a delay that is an integer multiple of the symbol time, and compensate the values at the ideal sampling point. This capability is sufficient for synchronized single-user MIMO since only these ideal sample points are useful for decoding.

However, for IN, when signals of multiple transmitters are misaligned, a symbol-rate equalizer is not sufficient. Figure 9 illustrates this issue. In Figure 9, we transmit a single frame and perform IN based on the two received signals. Since there is only one frame, the result of IN should be pure noise. Figure 9(a) shows the energy of the original signal and (b) shows the results of IN. We can see that with a conventional symbol-rate equalizer, peaks of residual energy remain which are due to the multi-path signals with fractional delays. In overlapped transmission, since the symbol timing is mis-aligned such residual energy will still cause interference to the remaining frame if they happen to overlap at the ideal sampling points.

To address this problem, in SAM we use an equalizer that works with over-sampling. Currently, we use four-time over-sampling in our system. Such an equalizer is usually referred to as a *fractional spaced equalizer* (FSE) [15]. However, unlike a conventional FSE which only trains an equalizer to minimize the mean square error at the ideal sampling points, in IN, the equalizer is trained to minimize the mean square error of all over-sampling points.

Such an approach immediately raises the following issue. The transmitter only specifies the value at the sampling point. What should be the expected values at the over-sampling points? Fortunately, we can get these expected values using interpolation from its nearby sample point values. For example, in 802.11b all transmitted samples are shaped with a root-raised cosine function. Thus, we can interpolate the fractional point values using the square of the power-shaping filter as

$$\tilde{x}(nT + \tau) = \sum_{i=-L}^{L} x(nT - iT) R(nT + \tau - iT),$$

where T is the symbol interval, $R(.)$ is the root-raised cosine function, and τ is the fraction of sampling time. When using four-time over-sampling, τ can be $\frac{T}{4}$, $\frac{T}{2}$, and $\frac{3T}{4}$.

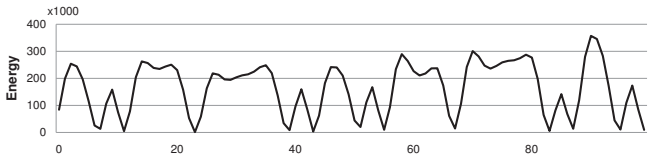
The algorithm outlined in Figure 8(a) can still be applied without modification, as long as the input of x_i and y_i are all over-sampling values, and each tap of $\boldsymbol{c}$ represents the coefficient on an over-sampling point. At each iteration, an estimation of error between the desired value and the recovered value is calculated. We use this error to drive the coefficient vector in the gradient direction with a step size controlled by Δ (line 10). When the searching converges, $\boldsymbol{c} = \{c_j\}$ is the optimum equalizer we need.

During our implementation, however, we found that the basic algorithm in Figure 8(a) is sensitive to the input data and may be unstable when some samples are extremely tiny (deep fading), which may cause an unreasonably large update in the coefficients. Although the situation could be controlled with a very tiny Δ, the tradeoff is a correspondingly slower convergence speed. Thus, we modify the basic algorithm with a more stable version, as shown in Figure 8(b). Instead of adjusting for every sample, we calculate a weighted average over a block of K samples and use this average to update the coefficient vector. As shown by lines 7–16, we calculate an "averaged" estimation of the gradient as a result.

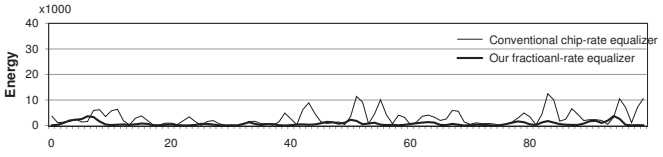
Figure 9(b) shows the results of our equalizer. After equalization, multi-path distortion can be effectively mitigated and over 90% of the original signal energy can be nullified.

(b) Decoding of remaining frame after IN

After IN, it is now straightforward to decode P_b. Since the interference of P_a has been removed, the preamble of P_b is now clean. So the receiver can re-synchronize to P_b, estimate the frequency offset and channel parameters and decode it using the normal decoder. Note that this wireless channel is actually a combined effect of the original wireless channels ($H_{i,j}, i, j = 1, 2$).



(a)



(b)

Figure 9: Equalization for interference nullifying in multi-path fading channel. IN is performed on a single frame transmitted to two receiving antennas. (a) Original interference signal. (b) The nullifying results with the conventional equalizer and our extended equalizer design.

3.2 Interference Cancelation

After P_b is decoded, it can be re-encoded and canceled from the original signals. To cancel P_b, the AP must regenerate a series of

samples, each of which has the exactly same phase shifting caused by the sample timing offset, carrier frequency offset, and the same distortion of the wireless channel. Assuming all this information is known, the generation of symbols of P_b at receiving antenna i takes the following steps:

1. **Re-generate timing offset.** We first need to generate the effect of the timing offset between the sender and receiver. Again, we use interpolation to obtain values at the receiver's sampling points, with

$$\tilde{y}_i(nT+\tau) = \sum_{j=-L}^{L} x(nT-jT)R(nT+\tau-jT),$$

where $R(.)$ is the root-raised cosine function and τ is the estimated sampling offset.

2. **Re-generate frequency offset.** Then, we add the phase shift caused by the frequency offset, using

$$\hat{y}_i[nT+\tau] = \tilde{y}_i(nT+\tau)e^{j2\pi\Delta f(nT+\tau)}.$$

3. **Re-generate channel distortion.** Finally, we pass the generated samples through a channel filter to add the multi-path distortions,

$$y_i[nT+\tau] = \boldsymbol{h}_{i2} * \hat{y}_i[nT+\tau],$$

where $\boldsymbol{h}_{i2}$ represents the channel vector between the i^{th} receiving antenna and the sending antenna of P_b.

Recall that, when decoding P_b, the receiver can already estimate the frequency offset and symbol timing of P_b, and the only unknown factor are the wireless channel coefficients. One challenge is that the preamble of P_b overlaps with P_a. Thus, as mentioned earlier, when purely relying on the preamble we cannot get a precise estimation of the channel coefficients. Fortunately, since we have already decoded P_b, we can use all of P_b as training symbols and thereby can greatly mitigate the impact of the interference from P_a.

Channel estimation is similar to finding an equalizer, but now we find a vector $\boldsymbol{h}$, such that

$$\boldsymbol{h} = \arg\min_{\boldsymbol{h}} ||\boldsymbol{y} - \boldsymbol{h} * \boldsymbol{x}||,$$

where $\boldsymbol{x}$ is the transmitted signal, $\boldsymbol{y}$ is the received signal, and "$*$" is the convolution operation. The same algorithm listed in Figure 8 can be used to estimate the channel, but now we need to exchange x and y.

After the signal of P_b is canceled, we can decode P_a. Since we have two versions of P_a from two receiving antennas, we can decode them in parallel and P_a is received if any one is decoded successfully.

3.3 Frequency offset and symbol timing tracking

In chain-decoding, with IN the receiver can get an almost interference-free preamble for each frame in an overlapped transmission. Thus, it is straightforward to estimate the symbol timing and frequency offset to each transmitter. However, due to the limited length of the preamble, the estimation may not be accurate enough for the entire transmission. The estimation error will accumulate with the duration of the transmission, and in the worst case, it will cause a decoding error in the latter portion of transmission.

To track these parameters for the entire transmission, chain-coding divides a long transmission into blocks and decode them one-by-one as shown in Figure 10. An arrival of a new frame always starts a new block, and a block should always contain whole preambles. After each block is decoded, the decoded symbols are used as training symbols to update the estimations of frequency offset and symbol timing for all transmitters. The size of the block depends on the accuracy of the estimations; it can be larger if the estimation has a high accuracy. In our experience, a block less than $3ms$ has been sufficient for accurately tracking these parameters.

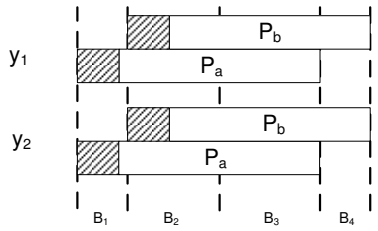


Figure 10: The tracking mode of chain-decoding. A long frame is divided into blocks. Chain-decoding decodes each block sequentially. The decoded symbols are then used for updating the symbol timing and frequency offset.

3.4 More than Two Frames Overlapped

It is straightforward to extend chain-decoding to N-antenna case. With N frames overlapped, the AP first uses the N received signals to nullify the first frame and get overlapped signals containing the $N-1$ frames. Then, we repeat the interference nullifying process to nullify proceeding frames. When only the last frame remains, the AP can decoded it and canceled it from the original frames. Then, recursively, all frames can be decoded. Note that the same procedures also apply to the tracking mode, where instead of the whole frame each block of the transmission is processed.

3.5 Discussion

SAM has the promise of linearly increasing of network capacity with the number of antennas at the AP by enabling spatial multi-access in WLANs. The major advantage of the chain-decoding method is that it does not require any strict synchronization between transmitters. It is also quite general because it works on various modulation schemes and with multi-carrier systems such as OFDM (although a better channel equalizer may be needed to take advantage of special OFDM structure).

There may still be some practical issues with SAM. First, although interference nullifying effectively removes the interference signal, it may reduce the signal strength of the desired signal if the wireless channels are destructive to each other. In particular, nullifying will have an adverse impact on decoding if it causes severe cancelation of the desired signal. Since the channel is randomly varying with time, the chance for such extreme case to happen is modest. As we will see later, SAM may only increase the frame loss rate slightly ($< 10\%$).

Second, interference nullifying and cancelation will also slightly increase the noise level in the resulted signal. Such increased noise will adversely affect the decoding reliability if the wireless link is at the low SNR boundaries. However, since WLAN is usually limited by interference instead of noise (high SNR) due to its short range, the slightly increased noise may not offset the gain of spatial multiplexing when only modest number of frames are overlapped. In Section 6, we show in with two antennas, SAM can substantially increase the network throughput in most cases.

Third, it is possible that the overlapped transmissions have such a large power difference that the higher power signal saturates the

receiving radio and the low power signal is lost. In our current system, we use 14-bit A/D convertors to provide over $30dB$ dynamics. Future APs may adopt even higher-end A/Ds to increase their dynamic range to better support spatial multiplexing. In addition, power-control algorithms can also be applied to coordinate the transmission power to fit into the receiving radio's dynamic range. We leave the study of power control in SAM as future work.

Finally, since SAM uses IN and IC sequentially to decode multiple concurrent frames, it may incur additional decoding latency compared to conventional single antenna system. However, since SAM uses a block-based decoding strategy (Section 3.3), it is possible to exploit pipeline parallelism to chain-decode. The latency can be controllable with a proper block-size.

4. CARRIER COUNTING MULTIPLE ACCESS

In addition to chain-decoding overlapped transmissions, spatial multi-access also requires stations to coordinate their transmissions for chain-decoding to be possible. Fortunately, SAM stations can use a simple distributed MAC for such coordination. This new MAC, named Carrier Counting Multiple Access (CCMA), is an extended version of the CSMA mechanism used in IEEE 802.11 DCF. CCMA is backward compatible to CSMA and can be incrementally deployed in current 802.11 networks.

4.1 Basic Idea

The basic idea behind the design of CCMA is the observation that when a preamble overlaps with other concurrent transmissions, though difficult to be used for parameter estimations, a receiver can still be able to reliably detect it. The reason is because preambles are fixed and usually transmitted using a robust modulation mode (*e.g.* 1Mbps in 802.11b). Figure 11 shows such an example. We have measured the transmission of two overlapped frames in IEEE 802.11b with 1Mbps modulation. To detect the preamble, we use the first $20\mu s$ (the length of a sensing slot in 802.11b) samples of a preamble as a template. Figure 11 shows the cross-correlation results. We can clearly see two peaks which indicate the start of two frames. Later in Section 6.2 we show that this correlation-based mechanism is a reliable technique, even in the presence of strong interference.

This observation implies that a station can learn the number of concurrent transmissions by observing the channel and detecting and counting preambles. Assume an AP has N antennas, thereby allowing up to N concurrent transmissions from different stations to the AP. A station with a backlogged frame for the AP will count the number of concurrent transmissions in the channel, say k, by sensing and counting preambles. If k is less than N, then there is a transmission opportunity for the station. We call this scheme *Carrier Counting Multiple Access* (CCMA).[2] Viewed from another perspective, CSMA may be regarded as a special case of CCMA where only one transmission is allowed in the wireless channel.

4.2 CCMA MAC Protocol

In this section, we assume all stations and the AP support CCMA. Later, we discuss backward compatibility where some stations or the AP are standard 802.11 devices.

We define *access threshold* K as the maximum number of concurrent transmissions allowed by the AP. The access threshold depends on the number of antennas, N, that the AP has. Normally, we have $K \leq N$. A SAM-enabled AP will announce its access threshold in the beacon message, and a station will record the threshold when it first associates with the AP.

[2]Although CCMA technically counts preambles, not only the carrier, we preferred this name since it a straightforward extension of CSMA for supporting spatial multiple access.

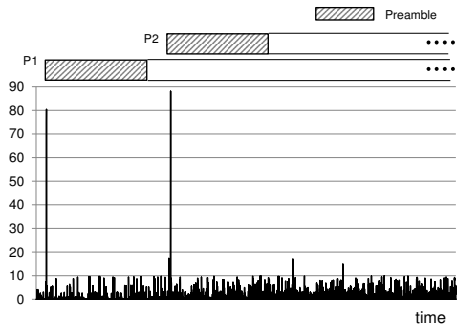


Figure 11: Detection of a preamble. The cross-correlation result of two measured overlapped frames transmitted at 1MBs with a $20\mu s$ sample template of preamble.

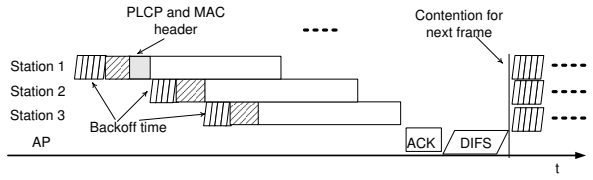


Figure 12: Transmission bursts for CCMA MAC protocol.

Like CSMA, CCMA is a pure contention-based protocol. Each station will maintain a counter for concurrent transmissions. When a station senses an idle channel (no transmission), it resets its counter to zero.

Assume at the beginning that the channel is idle. The backlogged stations contend for the first transmission opportunity, as shown in Figure 12. CCMA uses the same backoff procedure and contention window parameters as in IEEE 802.11 to provide backward compatibility. Once one station, say station 1, wins the contention, it will start transmitting its frame immediately. Now, all other stations should detect the preamble transmission of station 1, and increase their counters by one. Then, all stations continuously monitor the channel and decode the MAC header of station 1's frame, from which the stations can learn the transmission target. If the target is a SAM-enabled AP with an access threshold K larger than one, additional transmission opportunities remain for other backlogged stations to transmit to the same AP.

Similarly, all backlogged stations will contend for the second transmission with a random backoff. Again, the winning station, say station 2, starts its transmission and all other stations update their counters. All stations should pause for a fixed time duration larger than the length of a preamble. Then, they contend for the third transmission opportunity, if allowed by the AP's threshold K. Once the counter is equal or larger than K, the station should defer its contention attempt until the channel becomes idle again and the counter is reset to zero. Thus, up to K frames may be overlapped to form a transmission burst. At the end of transmission burst, an ACK-to-All is transmitted by the AP to acknowledge all received frames in the last burst. Then, after the channel becomes idle again for a certain period of time, all stations start to contention for transmission of next frames, as shown in Figure 12.

A collision could happen when two stations try to send in the same contention time slot. Collisions may result in decoding failure during chain-decoding, since two preambles overlap together and may cause unreliable parameter estimation. In such cases, all transmitted stations may not receive an ACK-to-All message. When a collision is detected, it is crucial for all transmitting stations to backoff to prevent collision-collapse in a contention-based protocol. While the optimal backoff strategy for CCMA is subject to future work, we propose two possible methods to manage contention. First, we can use a similar exponential backoff strategy as in IEEE 802.11: when a collision happens, the contention window is doubled. Although this mechanism has proved to be effective in 802.11 to resolve collisions, it reduces the gain of SAM and CCMA since smaller portions of frames are overlapped. Alternatively, it is possible for all contending stations to pick a random number p such that each station will only contend for the next transmission opportunity with a probability of p. Note that these two methods can be combined to manage the contention in CCMA.

In addition, there are also a few points that should be considered:

(a) Selective acknowledgement. In current CCMA, one single ACK frame is used to acknowledge all frames in an overlapped transmission. This design is suitable when the major frame losses are caused by collisions, since when collision happens, all frames may fail to be decoded. When the major reason of frame losses are channel errors, it would be possible to extend the ACK-to-All frame to *Selective Acknowledgement* in CCMA. In this case, the ACK frame may contain multiple fields to indicate the senders of successfully received frames, so to avoid unnecessary retransmissions. A detailed study on the acknowledge strategy will be a future work.

(b) Busy channel without detecting a valid preamble. It is possible that a station senses transmission energy on the channel but fails to detect a preamble. For example, it may be a transmission from a non-SAM/802.11 device, or it may happen when a station returns to the receiving mode after its transmission and therefore misses the preamble of other concurrent transmissions. In either of these two cases, the station loses track of the channel and it should behave conservatively by deferring its transmission attempts until the channel is idle again.

(c) Hidden terminal. Based on sensing, CCMA shares the similar hidden terminal issue like CSMA. Although there have been extensive proposals to address the limitations of CSMA in the literature, the hidden terminal problem is still not well-solved. We defer the detailed study of CCMA with hidden terminals to future work. However, we would like to point that the impact of hidden terminal may be mitigated with Chain-decoding. A SAM-enabled AP can announce an access threshold K to be smaller than its actually deployed antennas. Thus, when frames from hidden terminals collide, the AP may still be able to decode them if their preambles do not collide. This is very likely as hidden terminals access channel in a completely random and asynchronous manner.

4.3 Backward compatibility

CCMA can be considered a superset of the conventional CSMA protocol. As a result, it is straightforward to be compatible with CSMA.

There are two situations to consider. First, if the AP is a legacy CSMA station, it will not announce an access threshold to CCMA stations. The CCMA stations will therefore use the default value of one and behave just like CSMA stations.

Second, if the AP uses CCMA, both it and associated CCMA stations will need to support and interact with legacy CSMA stations. When a SAM-enabled AP has announced an access threshold K, all CCMA and CSMA stations contend for the first transmission opportunity using the standard mechanism as defined by the 802.11 MAC.

If a CCMA station wins, then other CCMA stations may continue to contend for the second transmission opportunity, as described in Section 4.2. All CSMA stations will hold their transmission attempts until the channel is idle again.

If a CSMA station wins, then all stations including CCMA nodes and other CSMA nodes will also defer their transmissions so that they do not interfere with the anticipated subsequent ACK frame. To differentiate a CCMA transmission from a conventional CSMA frame, a reserved bit in the CCMA frame header (*i.e.* PLCP header in 802.11) can be used. When an AP receives a legacy frame from a CSMA station, it will return a normal ACK message.

All stations, including CCMA and CSMA stations, contend for the transmission opportunity fairly once the channel becomes idle. The overlapped CCMA transmissions may occupy channel longer as a latter frame overlaps the former frame only after its preamble. However, such difference should be modest. For example, if $K = 2$ and short preamble is used in 802.11b, the overlapped transmission is only 10% longer than a normal full size 802.11 frame. With $K = 5$, the overlapped transmission is roughly 50% larger. Thus, CCMA is fair to CSMA and the deployment of CCMA will not cause significant throughput degradation to normal CSMA nodes.

5. IMPLEMENTATION

We have implemented SAM using the Sora high-performance software radio platform [16]. By using multi-core CPUs, Sora allows all PHY and MAC processing in software on the host. As a result, it provides a convenient environment for the implementation and evaluation of experimental wireless LAN systems like SAM.

5.1 Sora Platform

Sora is a fully programmable software radio platform based on commodity general-purpose PC architectures. Sora includes a Radio Control Board (RCB), which can connect to various radio front-ends, and interfaces with the PC using the high-speed, low-latency PCIe bus. The current Sora RCB implements PCIe-x8 and achieves 12Gbps throughput, sufficient for transferring high-speed, wide-band digital samples between the RF front-end and PC memory. All PHY and MAC processing is done in software on the host CPU. A full-featured IEEE 802.11a/b/g PHY and CSMA MAC, called SoftWiFi, has been implemented entirely in software on Sora. SoftWiFi seamlessly inter-operates with commercial 802.11 NICs in all modulation rates, and achieves the equivalent performance.

5.2 SAM Implementation

We have implemented chain-decoding based on the 802.11b PHY in SoftWiFi that supports modulation rates of 1, 2, 5.5 and 11Mbps. We have further implemented CCMA based on SoftWiFi's CSMA MAC. The current CCMA implementation mimics the behavior of 802.11b stations when contending for transmissions. The initial backoff window is set to 8 and the backoff slot is $20\mu s$.

We have implemented a full chain-decoder that includes interference nullifying, cancelation, 11b demodulation, and tracking. At the time of writing this paper, our implementation has not been fully optimized to operate at the real-time speed. From our initial experience, we believe the time complexity of chain-decoding is a small linear factor of conventional 11b decoder that can run in real-

time on a 2.66GHz CPU core in Sora. Thus, the overhead should be manageable by exploiting the computational power of additional CPU cores in the near future.

Due to the current limited availability of RF front-ends, in our current environment we only have two antennas installed at the SAM AP. When the SAM AP detects a frame, the chain-decoder will first decode the signal from each antenna. If neither of the frames can be decoded, suggesting potential overlapped frames, the decoder will perform interference nullifying and cancelation as described in Section 3. If any frame is decoded successfully at the first stage of chain-decoding, the decoder will check if there are other frames overlapped; if true, interference cancelation is performed and the decoder tries to decode the overlapped frame.

6. EVALUATION

In this section, we evaluate SAM using Sora software radio testbed. Our goal is to show SAM is plausible in practical wireless LAN environment. We conduct micro-benchmark to evaluate the three key techniques in SAM, including interference nullifying, cancelation and preamble sensing. We then show the benefit of spatial multi-access by measuring the end-to-end throughput gain of SAM over commodity 802.11b network.

6.1 Experimental setup

Figure 13 shows the layout of our testing environment. It is a typical office floor with cubicles separated by high walls. The AP is located at the center of the floor. We use two Sora nodes each connects to an antenna to form a two-antenna AP. Another two Sora nodes running CCMA protocol and laptops equipped with commodity 802.11 card (based on Atheros 5212 chipset) are moved among the five locations during our experiments, as shown in Figure 13. All mobile stations are configured to run 802.11b PHY layer and the AP runs chain-decoding algorithm. In following experiments, we mainly focus on the high SNR wireless links with high modulation rate (*i.e.* 5.5 and 11Mbps).

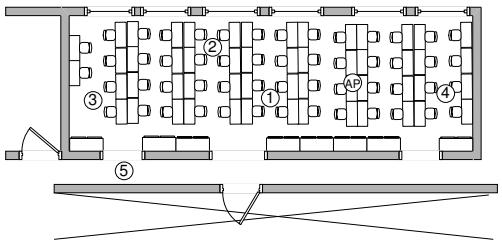


Figure 13: SAM test environment.

6.2 Micro-benchmark

(a) Interference Nullifying

We first evaluate the performance of interference nullifying in the chain-decoder implementation. We randomly place the CCMA nodes in the five locations in Figure 13. We also adjust the power level to generate different SNR, but in each setting, we ensure the wireless link is able to reliably receive non-overlapped frames (frame loss < 10%). We instruct CCMA nodes to transmit 1024-byte UDP frames modulated in 5.5Mbps rate. We dump the data at the AP side of 10,000 overlapped frames. We perform the IN algorithm on the dumped data. To evaluate how effective our IN algorithm performs, we examine the Residual Signal-plus-Noise-to-Noise Ratio (RSNR) after IN under different SNR settings. RSNR is defined as following

$$\frac{R+N}{N},$$

where R is the residual energy signal of the first frame after IN and N is the energy of noise. RSNR can be regarded as the ratio of noise that is enlarged by IN.

Figure 14 shows the result. The x-axis shows the sum of SNR values of the first frame at both antennas. We can make the following observations. First, the majority frames are with SNR between 9dB to 20dB. This is normal because that is the usual SNR range for 5.5Mbps 11b to work reliably in practice. However, we can still observe frames with very low SNRs. This is due to the dynamic fading effects of wireless channel. Second, we can see that IN has removed a significant portion of interfering energy, and the majority of RSNR is very small, with 0 to 2dB higher than the noise (or in other words, increasing the noise only by less than 58%). Such increased noise may affect the decoding reliability if the wireless link works at exactly boundary SNR, *i.e.* 6dB, which is the minimal required SNR for 5.5Mbps frames. However, in practice, many wireless links work at higher SNR range than this minimal requirement. For example, in our data set, most links have a SNR higher than 9dB. Thus, this additional noise has little impacts on the following decoding process. Finally, we can observe in all SNR settings, there are sparse points where the RSNR has almost the same value of SNR. This means our IN algorithm fails to nullify the interference. This is because these frames experienced severe multipath fading which distorts our channel estimation. Figure 15 plots the Cumulative Distribution Function (CDF) of the RSNR values. We can observe that over 90% cases, IN can effectively remove the interference and the resulted RSNR is less than 2dB. There are only around 5% cases that RSNR is larger than 3dB where the decoding of the second frame may be affected.

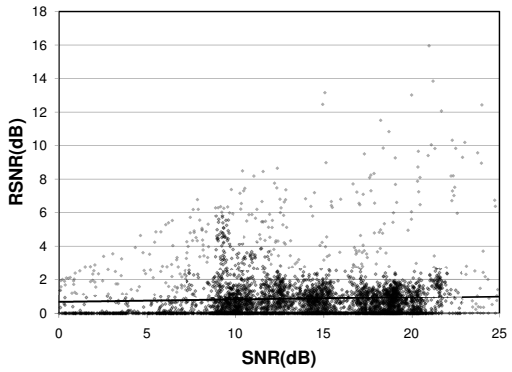


Figure 14: Performance of interference nullifying. The solid line plots the linear regression of the data.

(b) Interference Cancelation

Next, we evaluate the performance of interference cancelation in the chain-decoding implementation. We use the same data as in the previous experiment. We first remove the cases where the second frame fails to decode even after interference nullifying (about 3%), since in this case the chain-decoding has to stop. We then remove the cases where the first frame can be decoded without further cancelation, or the first frame captures. This portion is about 19%. Then, we evaluate how effective our IC algorithm can improve the signal-to-interference-and-noise (SINR) value of the first

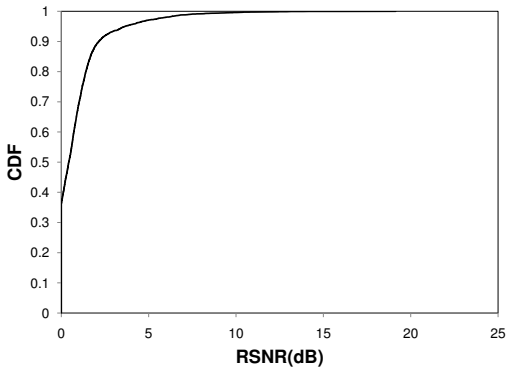


Figure 15: CDF of RSNR of interference nullifying.

frame and thereby enable correctly decoding. SINR is defined as:

$$SINR = \frac{S}{I+N},$$

where S is the signal energy of the first frame, I is the energy of interference frames, and N is the noise.

Figure 16 shows the SINR of the first frame after canceling the interference from the second frame. The x-axis is the SINR before IC. The solid-line shows the "y=x" line, and therefore the distance to that line actually shows the improved SINR (in dB) by IC. We make following observations. First, IC effectively improves SINR of the frame by reducing the energy from interference. From Figure 16, we can see that SINR has been substantially improved to 6–17dB, thereby these undecodable frames can all be virtually decoded successfully. Second, the improved SINR value reduces as the original SINR increases. This is reasonable since SINR is a ratio between the desired signal energy and the sum of interference and noise. A larger SINR usually means the interference is small. Thus, cancelation of the interference may have less impact on SINR since the term N may dominate. Lastly, we note that these portion of frames (78% of our data set) can only be decoded using multi-antenna systems like SAM with both IN and IC. They cannot be decoded using IC only in single antenna systems since no capture occurs.

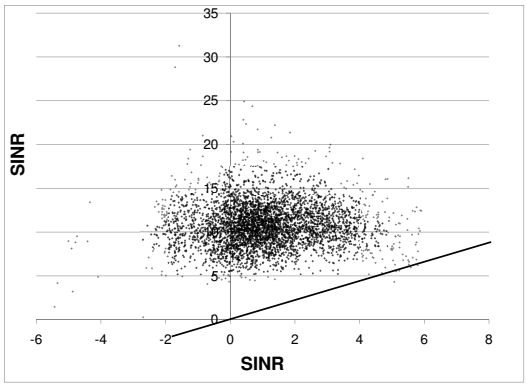


Figure 16: Performance of interference cancelation. The x-axis shows SINR of the first frame before IC; the y-axis shows SINR after IC. The solid line shows the "y=x" line.

(c) Preamble Sensing

We evaluate the correlation-based preamble sensing algorithm in CCMA under different SINR settings. We use the first $20\mu s$ samples of a preamble as the correlation template. We generate various number of overlapping 802.11b frames (from 2 to 10) using Sora[3]. We run the algorithm over 1500 such frames to detect each preamble in overlapped transmissions.

Figure 17 shows the results. False positives correspond to the probability that the algorithm falsely detects a preamble when one does not exist. False negatives correspond to the probability that the algorithm misses a real preamble and report nothing. From Figure 17, we can observe that as SINR decreases, false positives also slightly increase to about 21% when SINR is as low as $-10dB$. This behavior is reasonable because when the interference and noise grow stronger, the correlation operation tends to give a higher value that triggers a false alarm. A false positive detection causes a sender to mistakenly assume a transmission has happened and therefore increase its counter. As a result, it may waste a transmission opportunity. But we can see, even with a very low SINR, the false positive rate is still low. Therefore, the impact on the throughput gain of SAM may be limited.

The false negative rate, on the other hand, is always negligible, even in very low SINR. This result means CCMA will reliably detect a preamble. Thus, once a frame starts transmitting, all nearby stations will reliably detect it and hold their transmissions to avoid collision in preambles.

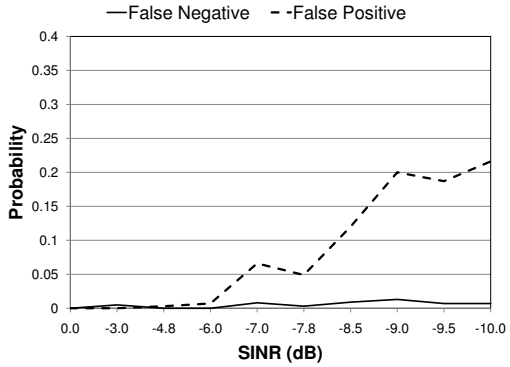


Figure 17: Performance of preamble sensing algorithm in CCMA.

6.3 Macro-benchmark

In this section, we evaluate the end-to-end throughput gain of SAM under the testbed shown in Figure 13. We mainly compare the throughput of SAM to the commodity 802.11b network. To measure 802.11b, we simply generate 1024-byte UDP packets using a laptop equipped with a commercial NIC and broadcast to the SAM AP. We record the throughput we get using SoftWiFi receiver [16]. However, it is slightly tricky to measure SAM's performance since our current chain-decoder cannot run in real-time. In this paper, we use the following method to measure the throughput of SAM. We simultaneously generate two 1024-byte UDP packet streams on two CCMA nodes and broadcast. The AP will take

[3]Due to the limited number of our hardware, we cannot actually create 10 overlapped frames. We use 3 Sora nodes to create overlapped transmissions. To emulate more overlaps, we first mix a few frames at baseband before transmitting using Sora.

a snapshot of the channel every 10 seconds. The snapshot contains around $100ms$ of four-time over-sampled signal data on one 20MHz 802.11b channel. We store the snapshot data on disk and process it at offline. Each snapshot contains around 40–70 frames when they are modulated using the 5.5 or 11Mbps rates. With these traces we can therefore estimate the throughput over the duration of a snapshot. The data presented is an average of all snapshots over 20 minutes. In the experiment, we ensure that all links can reliably support 5.5Mbps or 11Mbps modulation rate.

Figure 18 shows the throughput for both commodity 802.11b nodes and SAM nodes at each location. We can observe that for all cases, SAM significantly improves the network throughput compared to 802.11b. It shows that spatial multi-access can make full use of two antennas at the AP by overlapping two concurrent transmissions from mobile stations. The throughput improvement of SAM over 802.11b ranges from 31–61% at 11Mbps, and from 45–76% at 5.5Mbps. Note that with higher modulation rates, the benefits of SAM slightly degrades. This degradation is due to the change in durations of the preamble relative to the frame duration at different rates. In SAM concurrent frames are partially overlapped, where subsequent frames in a burst start after the preambles of previous frames. As the modulation rate increases, the duration of a frame transmission becomes shorter while the preamble duration remains fixed. Consequently, the relative overhead of the preamble increases.

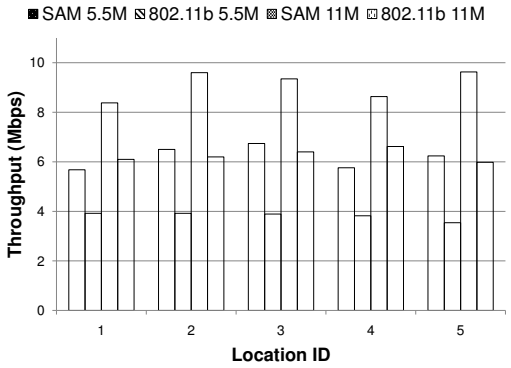


Figure 18: End-to-end throughput.

Table 1 also summarizes the frame loss rate for both 802.11 and SAM at different locations. As expected, SAM slightly increases the frame loss rate in tradeoff to being able to transmit frames concurrently. Given the relatively small increase in frame drop rate in light of the substantial increase in throughput, we consider this a reasonable tradeoff.

Table 1: Frame loss rate.

Location		1	2	3	4	5
5.5M	802.11	2.7%	2.9%	3.6%	5.3%	11.0%
	SAM	10.3%	3.7%	4.8%	5.6%	9.7%
11M	802.11	2.1%	5.9%	4.0%	3.4%	8.5%
	SAM	5.8%	6.5%	5.5%	9.3%	9.9%

7. RELATED WORK

Our work is based on the theory foundation of MIMO and its recent advance in multi-user MIMO (MU-MIMO) [2, 5, 9, 14, 17]. Conventional single user MIMO systems, *e.g.* 802.11n [1] and BLAST [4, 20], support only point-to-point communication, where the sampling clocks for all antennas at the transmitter (or receiver) are completely synchronized. With single-user MIMO, the capacity improvement is bounded by the number of transmitter or receiver antennas, whichever is smaller. Therefore, in practice, the number of antennas at the mobile station usually constrains the network capacity. In contrast, multi-user MIMO (MU-MIMO) systems allow multiple stations to transmit concurrently thereby fully utilizing the AP's antennas [2, 5, 9]. However, most MU-MIMO systems are based on cellular networks, where the central base station strictly controls and tightly synchronizes all mobile stations. This way, the central base station can precisely measure the spatial signatures, including the wireless channels and the delay of the radio signal from sending antenna to each receiving antenna, and with such information, data streams from different users can be separated using a linear decorrelator or MMSE-SIC algorithms [17,18]. In contrast, SAM uses a cross-layer approach with PHY and MAC techniques that target the distributed, uncontrolled environment. To the best of our knowledge, this paper is the first to present a system design and an implementation of a spatial multi-access in uncontrolled wireless LAN.

Our work extensively uses the advanced signal processing techniques like interference nullifying and cancelation. Such ideas have been well studied in communication theory. Recently, systems based on interference cancelation also demonstrate its application in many scenarios. ZigZag [6] recursively applies IC to decode N frames from N collisions. Although ZigZag is a clever way to resolve collisions, it does not increase the wireless network capacity. Halperin *et. al.* [8] shows a practical implementation of IC to decode concurrent frames. However, as it works in a single antenna system, it requires one of the concurrently transmitted frames to be much stronger than the other frames so that it can be captured. As the transmission rate increases, frames require a higher SINR to be captured. Therefore, a high-rate frame can be easily overwhelmed by other transmissions when stations do not carefully control their transmission power. SAM also combines interference nullifying and cancelation to enable concurrent transmission. Unlike SIC, SAM does not have such constraints on transmission rate and power. High-rate frames with similar power can still be reliably decoded as they can be separated along the spatial dimension. Further, from an information theory point of view, SIC can only improve network capacity logarithmically through power control; while by exploiting the spatial dimension, SAM can potentially increase network capacity nearly linearly.

One alternative way to use multiple antennas is to improve the reliability of single transmission by exploiting the spatial diversity. For example, it can combine the signals received from multiple antennas based on Maximal Ratio Combination (MRC) [15, 17]. MRC may increase effective SNR and therefore the wireless capacity, since one could transmit using a higher modulation rate. However, the result from Information Theory has shown that the capacity only increases logarithmically with SNR. Thus, in high SNR environment like WLAN, spatial multiplexing schemes like SAM can improve the wireless capacity much substantially [17].

Spatial diversity can also be exploited with collaboration among densely deployed APs [12, 13, 21]. However, these approaches can only improve the reliability of a single frame transmission; while SAM enables spatial multi-access and improves network capacity by allowing concurrent transmissions.

There are also other MAC layer techniques that can increase wireless spatial reuse [10, 19]. These approaches increase concurrency of transmissions only when they do not interfere with one another. Compared to these MAC solutions, SAM increases transmission concurrency by allowing stations to interfere with each other with overlapped transmissions (CCMA). With chain-decoding, the

AP can still decode these overlapped frames using signals received from multiple antennas.

There is a recent work, IAC, that improves the capacity in MIMO-based wireless LAN where AP has only limited antennas and thus the bottleneck for further capacity improvement [7]. IAC allows multiple client-AP pairs to transmit concurrently by interference alignment and cancelation with collaborative APs. However, we believe it may be more likely the mobile station is the limiting factor than AP. Thus, in this paper, we addresses the issue to make full use of AP's antennas by enabling multiple concurrent access of mobile stations.

8. CONCLUSION

This paper presents SAM, a system for supporting spatial multiple access in wireless LANs to increase network capacity. SAM uses a cross-layer design to address both PHY and MAC challenges due to the asynchronous and uncoordinated nature of WLANs. SAM combines two new techniques to make spatial multi-access feasible and practical to implement in typical WLAN environments. First, SAM uses *chain-decoding* to reliably decode concurrent frame transmissions which are partially overlapped in a cascade staggered on frame preambles. With this approach, chain-decoding can reliably estimate system parameters for each station and repeat *interference nullifying and cancelation* to decode all overlapped frames. Second, we also proposed a new distributed MAC layer protocol, *Carrier Counting Multiple Access* (CCMA), to enable concurrent transmissions in a random access WLAN. CCMA enables WLANs to retain their asynchronous nature while supporting spatial multi-access. CCMA also supports mixed networks with CSMA stations for backwards compatibility.

We have implemented SAM on the Sora high-performance software radio platform using commercially-compatible implementations of the IEEE 802.11b PHY and MAC. We demonstrate that spatial multi-access with SAM has a throughput gain of 45–76% with a 5.5Mbps modulation rate, and 31–61% with 11Mbps, relative to standard CSMA with IEEE 802.11b. We note that the design of chain-decoding is not limited to just the 802.11b PHY. It is a more general technique that can be applied to other modulation algorithms, *e.g.* OFDM.

Currently, our chain-decoder implementation is still not fast enough for real-time processing of 802.11 signals, and our performance evaluation relies upon an offline decoder. However, as SAM introduces only a linear-time complexity increase with the number of antennas at the AP, we see no obstacle that would prevent a SAM implementation from running in real-time on a commodity PC after careful optimization. We are planning to develop an online version of SAM in the near future.

Our work may also be extended to enable spatial multiplexing in downlink. From Information Theory, it is possible for multi-antenna AP to pre-code multiple frames to reverse the channel effects and simultaneously transmit to multiple mobile stations. Thus, each mobile station will receive signal only for itself while other interferences are nullified [15]. However, such pre-coding schemes are general complex and challenging to be implemented efficiently in practice. Besides, it is also challenging for AP to reliably learn wireless channels to the mobile stations in wide-band dynamic wireless systems. How to address these challenges is our future work.

9. ACKNOWLEDGEMENTS

The authors would like to thank Hui Liu from University of Washington, WA, USA, for his valuable comments and suggestions on our work.

10. REFERENCES

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

[2] 3GPP. UL virtual MIMO transmission for E-UTRA. *3GPP TSG-RAN1 #42bis*, 2005.

[3] D. G. Brennan. On the maximal signal-to-noise ratio realizable from several noisy signals. In *IRE*, 1955s.

[4] G. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multiple antennas. In *Bell Labs, Technichal Journal*, 1996.

[5] D. Gesbert, M. Kountouris, R. W. Heath, C. byoung Chae, and T. Salzer. From single user to multiuser communications: Shifting the mimo paradigm. In *IEEE Sig. Proc. Magazine*, 2007.

[6] S. Gollakota and D. Katabi. Zigzag decoding: Combating hidden terminals in wireless networks. In *SIGCOMM '08*, 2008.

[7] S. Gollakota, S. D. Perli, and D. Katabi. Interference alignment and cancellation. In *SIGCOMM '09*, 2009.

[8] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *Proceedings of ACM MobiCom*, 2008.

[9] M. Jiang and L. Hanzo. Multiuser MIMO-OFDM for next-generation wireless systems. *Proceedings of the IEEE*, 95(7):1430–1469, 2007.

[10] T.-S. Kim, H. Lim, and J. C. Hou. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *ACM MobiCom '06*, 2006.

[11] U. Mengali and A. N. D. Andrea. *Synchronization Techniques for Digital Receivers*. Cambridge University Press, 1997.

[12] A. Miu, G. Tan, H. Balakrishnan, and J. Apostolopoulos. Divert: Fine-grained path selection for wireless LANs. In *MobiSys'04*, 2004.

[13] A. K. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *11th ACM MOBICOM Conference*, Cologne, Germany, September 2005.

[14] C. Oestges and B. Clerckx. *MIMO Wireless Communications: From Real-World Propagation to Space-Time Code Design*. Academic Press, Inc., Orlando, FL, USA, 2007.

[15] J. G. Proakis and M. Salehi. *Digital Communications*. McGraw Hill, 2008.

[16] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker. Sora: High performance software radio using general purpose multi-core processors. In *NSDI 2009*.

[17] D. Tse and P. Vishwanath. *Fundamentals of Wireless Communications*. Plenum Press New York and London, 2005.

[18] S. Verdu. *Multiuser detection*. Cambridge Univ. Press, 1998.

[19] M. Vutukuru, K. Jamieson, and H. Balakrishnan. Harnessing Exposed Terminals in Wireless Networks. In *5th USENIX Symposium on Networked Systems Design and Implementation*, San Francisco, CA, April 2008.

[20] P. Wolniansky, G. Foschini, G. Golden, and R. Valenzuela. V-BLAST: an architecture for realizing very high data rates overthe rich-scattering wireless channel. In *URSI International Symposium on Signals, Systems, and Electronics, ISSSE 98*, 1998.

[21] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi. Beyond the bits: cooperative packet recovery using physical layer information. In *13th ACM MOBICOM Conference*, New York, NY, USA, 2007. ACM.