Network Multiple-Input and Multiple-Output for Wireless Local Area Networks

Hong Huang, Hajar Barani, and Hussein Al-Azzawi

Klipsch School of Electrical and Computer Engineering, New Mexico State University, NM, USA

Abstract—This paper presents a tutorial for network multiple-input and multiple-output (netMIMO) in wireless local area networks (WLAN). Wireless traffic demand is growing exponentially. NetMIMO allows access points (APs) in a WLAN to cooperate in their transmissions as if the APs form a single virtual MIMO node. NetMIMO can significantly increase network capacity by reducing interferences and contentions through the cooperation of the APs. This paper covers a few representative netMIMO methods, ranging from interference alignment and cancelation, channel access protocol to allow MIMO nodes to join ongoing transmissions, distributed synchronization, to interference and contention mitigation in multiple contention domains. We believe the netMIMO methods described here are just the beginning of the new technologies to address the challenge of ever-increasing wireless traffic demand, and the future will see even more new developments in this field.

Index Terms—MIMO, WLAN, cooperative radio transmission.

I. INTRODUCTION

Recently, there is an explosive growth of mobile traffic demand [6]. A number of technologies have been developed to respond to the traffic demand growth, such as network MIMO (netMIMO) in Wireless LAN (WLAN) [8], relay-based cooperative techniques in general wireless networks [14], femtocells and multi-user MIMO (MU-MIMO) in cell phone networks [7], [23], [24], etc. In this paper, we provide a tutorial on Network MIMO. In netMIMO, also called distributed MU-MIMO, APs in WLAN synchronize and collaborate in the transmissions of data packets as if the APs form a single virtual MIMO node. NetMIMO can potentially significantly increase network capacity by reducing interferences and contentions through the cooperation of the APs. The intended reader of this tutorial should have some basic backgrounds in wireless communications. Our intention is to enable the reader to implement and/or improve the netMIMO methods after reading this tutorial. Therefore, instead of covering a wide-range of methods, we focus on a few representative methods and provide very detailed description for the covered methods. Mathematical formulas are used liberally where they express the ideas most accurately. We hope this tutorial will be useful for the reader who has to deal with the challenge of the ever-increasing wireless traffic demand. We believe the netMIMO methods described here are just the beginning of the new technologies to address the challenge of wireless traffic demand, and the future will see even more new developments in this field.

A key attribute of netMIMO is its scalability, which we provide an overview below. Ideally, netMIMO allows the APs to transmit concurrently and enables the downlink capacity scale linearly with the number of APs in the network. In practice, scalability is hard to achieve, especially in large networks. There are five factors affecting the scalability of netMIMO in large networks: multiplexing gain, diversity gain, spatial reuse, and scalability in one contention domain and in multiple contention domains. In [29], the scalability of various WLAN architectures is compared, which is summarized in Table I and described in the following.

- SISO (single-input and single-output): In IEEE 802.11a/b/g SISO WLAN, neighboring APs interfere with each other, and only one neighbor can transmit at a time. Thus, there is no multiplexing gain. However, APs in different contention domains can have spatial reuse, leading to scalability in multiple contention domains.
- MIMO: In IEEE 802.11n MIMO WLAN, we can send multiple streams of data from the transmitter to the receiver, or beamform the data using multiple antennas to improve signal to interference and noise ratio (SINR) and link capacity, which brings diversity gain. However, like SISO, neighboring MIMO APs interfere with each other and there is no cooperation among them. Thus, there is no multiplexing gain. But, there is spatial reuse and scalability in multiple contention domains.
- MU-MIMO/CAS: In IEEE 802.11ac MU-MIMO WLAN [3], [18], [25], [26], [29], an AP with multiple antennas sends different data streams to multiple clients in the cell. Such a co-located antenna system (CAS) behaves much like MIMO: it provides diversity gain and scalability in multiple contention domains, but it does not provide multiplexing gain.
- NetMIMO: In netMIMO WLAN [4], [9], [16], [22], multiple APs that are tightly synchronized cooperate to send multiple data streams to their clients. These APs in effect form a "giant" multi-antenna transmitter. The APs transmit to their clients simultaneously by canceling out the interferences. Thus, there is multiplexing gain and scalability within the contention domain. However, such netMIMO schemes require synchronization in transmission time, sampling clock rate, carrier frequency and phase. They also require data sharing among all participating APs. Thus, these netMIMO schemes are not scalable with multiple contention domains.
- NEMOx: In NEMOx WLAN [29], a netMIMO variant that makes AP’s downlink capacity to scale as AP density and network size increase.

Abstract—This paper presents a tutorial for network multiple-input and multiple-output (netMIMO) in wireless local area networks (WLAN). Wireless traffic demand is growing exponentially. NetMIMO allows access points (APs) in a WLAN to cooperate in their transmissions as if the APs form a single virtual MIMO node. NetMIMO can significantly increase network capacity by reducing interferences and contentions through the cooperation of the APs. This paper covers a few representative netMIMO methods, ranging from interference alignment and cancelation, channel access protocol to allow MIMO nodes to join ongoing transmissions, distributed synchronization, to interference and contention mitigation in multiple contention domains. We believe the netMIMO methods described here are just the beginning of the new technologies to address the challenge of ever-increasing wireless traffic demand, and the future will see even more new developments in this field.

Index Terms—MIMO, WLAN, cooperative radio transmission.

I. INTRODUCTION

Recently, there is an explosive growth of mobile traffic demand [6]. A number of technologies have been developed to respond to the traffic demand growth, such as network MIMO (netMIMO) in Wireless LAN (WLAN) [8], relay-based cooperative techniques in general wireless networks [14], femtocells and multi-user MIMO (MU-MIMO) in cell phone networks [7], [23], [24], etc. In this paper, we provide a tutorial on Network MIMO. In netMIMO, also called distributed MU-MIMO, APs in WLAN synchronize and collaborate in the transmissions of data packets as if the APs form a single virtual MIMO node. NetMIMO can potentially significantly increase network capacity by reducing interferences and contentions through the cooperation of the APs. The intended reader of this tutorial should have some basic backgrounds in wireless communications. Our intention is to enable the reader to implement and/or improve the netMIMO methods after reading this tutorial. Therefore, instead of covering a wide-range of methods, we focus on a few representative methods and provide very detailed description for the covered methods. Mathematical formulas are used liberally where they express the ideas most accurately. We hope this tutorial will be useful for the reader who has to deal with the challenge of the ever-increasing wireless traffic demand. We believe the netMIMO methods described here are just the beginning of the new technologies to address the challenge of wireless traffic demand, and the future will see even more new developments in this field.

A key attribute of netMIMO is its scalability, which we provide an overview below. Ideally, netMIMO allows the APs to transmit concurrently and enables the downlink capacity scale linearly with the number of APs in the network. In practice, scalability is hard to achieve, especially in large networks. There are five factors affecting the scalability of netMIMO in large networks: multiplexing gain, diversity gain, spatial reuse, and scalability in one contention domain and in multiple contention domains. In [29], the scalability of various WLAN architectures is compared, which is summarized in Table I and described in the following.

- SISO (single-input and single-output): In IEEE 802.11a/b/g SISO WLAN, neighboring APs interfere with each other, and only one neighbor can transmit at a time. Thus, there is no multiplexing gain. However, APs in different contention domains can have spatial reuse, leading to scalability in multiple contention domains.
- MIMO: In IEEE 802.11n MIMO WLAN, we can send multiple streams of data from the transmitter to the receiver, or beamform the data using multiple antennas to improve signal to interference and noise ratio (SINR) and link capacity, which brings diversity gain. However, like SISO, neighboring MIMO APs interfere with each other and there is no cooperation among them. Thus, there is no multiplexing gain. But, there is spatial reuse and scalability in multiple contention domains.
- MU-MIMO/CAS: In IEEE 802.11ac MU-MIMO WLAN [3], [18], [25], [26], [29], an AP with multiple antennas sends different data streams to multiple clients in the cell. Such a co-located antenna system (CAS) behaves much like MIMO: it provides diversity gain and scalability in multiple contention domains, but it does not provide multiplexing gain.
- NetMIMO: In netMIMO WLAN [4], [9], [16], [22], multiple APs that are tightly synchronized cooperate to send multiple data streams to their clients. These APs in effect form a "giant" multi-antenna transmitter. The APs transmit to their clients simultaneously by canceling out the interferences. Thus, there is multiplexing gain and scalability within the contention domain. However, such netMIMO schemes require synchronization in transmission time, sampling clock rate, carrier frequency and phase. They also require data sharing among all participating APs. Thus, these netMIMO schemes are not scalable with multiple contention domains.
- NEMOx: In NEMOx WLAN [29], a netMIMO variant that makes AP’s downlink capacity to scale as AP density and network size increase.
The paper is organized as the following. In Section II, we describe a netMIMO scheme at the PHY layer that combines interference alignment and cancelation to increase multiplexing gain in a contention domain. In Section III, we describe a distributed channel access protocol that allows nodes with many antennas to join the existing transmissions without disrupting ongoing transmissions in a single contention domain. In Section IV, we describe a distributed scheme for synchronizing the phases of MIMO transmitters in a single contention domain. In Section V, we describe a netMIMO scheme that scales well in multiple contention domains.

II. INTERFERENCE ALIGNMENT AND CANCELLATION

In [9], a method for interference alignment and cancelation (IAC) is proposed. Combining interference alignment and cancelation applies to scenarios where neither interference alignment nor cancelation applies alone. In traditional WLANs, we use one AP to serve a particular contention domain. Adjacent contention domains use different channels to reduce interference. In IAC, adjacent contention domains still use different channels. However, different from traditional WLANs, there are multiple APs in a contention domain serving multiple clients simultaneously in IAC. In IAC, we assume the APs are connected via a wired infrastructure such as Ethernet, which is used to enable APs to collaborate with each other. One AP is selected as the master node that arranges the channel access and the process of interference alignment and cancelation. We also assume channel estimates are available, which can be obtained using standard techniques [2]. IAC operates using the existing modulation and coding schemes.

For clarity, we describe the scenario where there are two antennas per node, and assume the nodes already have the channel estimates. We also assume the interference is much stronger than noise and is the dominant factor in signal reception. In the following, we first describe how IAC works on the uplink with two, three, and four concurrent signals, respectively. Then, we describe how IAC works in the downlink. Finally, we cover the general case of an arbitrary number of antennas per node.

A. Two Concurrent Signals on the Uplink

In this scenario, a single client transmits two concurrent signals $x_1$ and $x_2$ to a single AP on the client’s first and second antennas, respectively. The channel linearly combines the two signals, and the received signals $y_1$ and $y_2$ at the AP are given by

$$
y_1 = h_{11}x_1 + h_{21}x_2
$$

$$
y_2 = h_{12}x_1 + h_{22}x_2
$$

where $h_{ij}$ is a complex number whose magnitude and angle represent the attenuation and the delay along the path from the client’s $i$th antenna to the AP’s $j$th antenna. The above equation can be written as

$$
y = Hx
$$

where $H = [h_{ij}]$ is the $2 \times 2$ uplink channel matrix. As shown in Figure 1, in the 2-dimensional vector space of the signal, the received signal at the AP is the sum of two vectors that are along the directions $H(1, 0)^T$ and $H(0, 1)^T$, respectively, where $T$ indicates matrix transpose. To decode $x_1$, the AP projects the received signal to the vector subspace orthogonal to $H(0, 1)^T$, which gets rid of the interference from $x_2$. Similar procedure is performed to decode $x_2$.

![Fig. 1. Two concurrent signals on the uplink.](image)

B. Three Concurrent Signals on the Uplink

In this scenario, there are two clients and two APs. Client 1 sends AP1 and AP2 signals $x_1$ and $x_2$, and client 2 sends AP1 signal $x_3$. So, there are three concurrent signals. The received signal at AP1 is given by

$$
y = H_{11}x_1 + H_{12}x_2 + H_{21}x_3
$$

where $H_{ij}$ are channel matrices from client 1 and client 2 to AP1. Since AP1 received only two signals $y_1$ and $y_2$, it can not decode three transmitted signals $x_1, x_2, x_3$. Similar situation is true for AP2.

To solve the above problem, we apply precoding to the transmitted signal. As shown in Figure 2, instead of transmitting $x_i$, we transmit $V_i x_i$, where $V_i$ is a 2-dimensional precoding vector. The precoding vectors are chosen so that the second and the third received signals are aligned, i.e.,

$$
H_{12}V_2 = H_{21}V_3.
$$

This can be achieved by picking random but unequal values for $V_1$ and $V_2$, and make $V_3 = H_{21}^{-1}H_{11}V_2$. Since the second...
the contribution from $x_1$ and $x_2$. With two received signals, AP3 can recover the remaining two signals $x_3$ and $x_4$. We can achieve the above alignment requirements by solving the following equations

alignment at AP1: $H_{11}V_2 = H_{21}V_3 = H_{31}V_4$ (4)

alignment at AP2: $H_{22}V_3 = H_{32}V_4$ (5)

where $H_{ij}$ is the channel matrix from client $i$ to AP $j$. In the above three linear vector equations, we have three unknown vectors $V_2, V_3, V_4$, which can be solved using standard techniques. Thus, by using the solutions as the precoding vectors $V_j, j = 2, 3, 4$, we can decode all four concurrent signals.

D. Downlink

The downlink is different from the uplink in that the clients cannot cooperate over the Ethernet. Suppose we have three APs and three clients. We want to deliver signals $x_1, x_2, x_3$ to clients 1, 2, 3. The AP$i$ transmits a signal $V_i x_i$. As shown in Figure 4, at client $j$, we decode $x_j$ by aligning the other two received signals and using orthogonal projection. Thus, we need to ensure

alignment at client 1: $H_{21}^d V_2 = H_{31}^d V_3$ (6)

alignment at client 2: $H_{12}^d V_1 = H_{32}^d V_3$ (7)

alignment at client 3: $H_{13}^d V_1 = H_{23}^d V_2$ (8)

where $H_{ij}^d$ is the channel matrix from $i$th AP to the $j$th client. The above three linear vector equations contain three unknown vectors $V_1, V_2, V_3$, which can be solved using standard techniques.
E. Beyond Two Antennas

For the general case of an arbitrary number of antennas per node, based on the fact that to have a feasible solution, the number of constraints imposed by interference alignment must be fewer than the number of concurrent signals (the unknowns), the following results are proven in [10]. For the downlink, in a system with $M$ antennas per node, the maximum number of concurrent signals IAC can deliver is $\max\{2M - 2, \lfloor 3M/2 \rfloor\}$. For $M > 2$, IAC achieves this with $M - 1$ APs. For the uplink, in a system with $M$ antennas per node and with three or more APs, the maximum number of concurrent signals IAC can deliver is $2M$.

IAC is experimentally validated in [9] in a 2x2 MIMO WLAN. The results show IAC increases the average throughput by 2x on the uplink and by 1.5x on the downlink.

III. MAC FOR HETEROGENEOUS NETMIMO

Today’s wireless devices have different form factors, ranging from desktopes and laptops, to tablets and smartphones, and to wireless sensors. The physical sizes of these devices put limits on the maximum number of antennas they can have. Thus, future wireless networks will be populated by heterogeneous devices having different numbers of antennas. Thus, there is need to design efficient MAC protocols for heterogenous netMIMO transmissions.

In [16], a distributed MAC protocol for netMIMO transmission, called 802.11n+, is proposed. In 802.11n+, nodes with different numbers of antennas contend for both time slots and degrees of freedom provided by multiple antennas. It is demonstrated that even when the medium is already occupied by some nodes, nodes with more antennas than the number of ongoing transmissions can transmit concurrently without interfering with the ongoing transmissions. Further, such nodes can contend for the medium in a distributed fashion.

In 802.11n+, a communication pair precedes its data exchange with a light-weight handshake, similar to but more efficient than RTS-CTS in 802.11. A transmitter that is about to join the ongoing transmissions uses the handshake message of prior contention winners to estimate the reverse channels from itself to receivers of the ongoing transmissions, using channel reciprocity. Channel reciprocity says that the forward and the reverse channels between the transmitter and the receiver have the same channel characteristics [11].

In the following, we first provide some illustrative examples of how 802.11n+ works. Then, we describe the 802.11n+ channel access protocol. Finally, we describe the 802.11n+ transmission protocol.

A. Illustrative Examples

In the following, we first describe two MIMO techniques we will use: interference nulling and interference alignment, assuming that the transmitter and receiver have the same number of antennas. Then, we cover the case where the transmitter and receiver have different numbers of antennas.

1) Interference Nulling: Consider a network with two pairs of transmitters (TX1, TX2) and receivers (RX1, RX2), referring to Figure 5. TX1 and RX1 have one antenna and start communications first. TX2 and RX2 want to join the communication. We prevent the transmissions at TX2 from interfering with the communications between TX1 and RX1 using a technique called interference nulling. Let $h_{ij}$ denote the channel coefficient from the $i$th antenna at the transmitter to the $j$th antenna at the receiver. To create a null at RX1, TX2 transmits a signal $x_2$ at its first antenna and $\alpha x_2$ at its second antenna. At RX1, the received signal is $(h_{21} + \alpha h_{31})x_2$. By choosing $\alpha = -h_{21}/h_{31}$, we ensure the signals from TX2’s two antennas cancel each other out, in effect nulling the interference.

Suppose the signal transmitted by TX1 is $x_1$. At RX2, the received signal is given by

$$y_2 = h_{12}x_1 + (h_{22} + \alpha h_{32})x_2 \tag{9}$$
$$y_3 = h_{13}x_1 + (h_{23} + \alpha h_{33})x_2 \tag{10}$$

Since we have two equations and two unknowns ($x_1, x_2$), we can decode $x_2$ at RX2.

2) Interference Alignment: Suppose we add a third communicating pair TX3 and RX3, both with 3 antennas to the network described in the previous section, referring to Figure 6. Since the network now has three degrees of freedom, we
To align the interference from TX1 and TX3, we select received signals at RX2 are given by RX2 sees only two signals, i.e., the signal transmitted by TX2 and the combined signal from TX1 and TX3. Thus, the received signals at RX2 are given by

\[
y_2 = h_{12} x_1 + (h_{22} + \alpha h_{32}) x_2 + (\alpha' h_{42} + \beta' h_{52} + \gamma' h_{62}) x_3
\]
\[
y_3 = h_{13} x_1 + (h_{23} + \alpha h_{33}) x_2 + (\alpha' h_{43} + \beta' h_{53} + \gamma' h_{63}) x_3.
\]

To align the interference from TX1 and TX3, we select \(\alpha', \beta'\) and \(\gamma'\) such that the following holds

\[
\frac{(\alpha' h_{42} + \beta' h_{52} + \gamma' h_{62})}{h_{12}} = \frac{(\alpha' h_{43} + \beta' h_{53} + \gamma' h_{63})}{h_{13}} = L
\]

where \(L\) is a constant. Using the above equation, we can rewrite the received signals at RX2 as

\[
y_2 = h_{12} (x_1 + L x_3) + (h_{22} + \alpha h_{32}) x_2
\]
\[
y_3 = h_{13} (x_1 + L x_3) + (h_{23} + \alpha h_{33}) x_2.
\]

Since RX2 has two equations and two unknowns \((x_2\) and \(x_3\)), it can decode the desired signal \(x_2\).

In sum, TX3 must satisfy two equations, i.e., the first equation in \((11)\) to null its interference at RX1, and \((13)\) to align its interference with that from TX1. This leaves the third degree of freedom for TX3 to transmit another signal to RX3.

We can continue with adding additional communicating pairs as long as they have additional antennas. By nulling at the first receiver and aligning at all the remaining receivers, each additional transmitter can send a signal to its receiver without interfering with the ongoing transmissions.

3) The Case Where Transmitter and Receiver Have Different Numbers of Antennas: Consider the case of two APs and three clients. AP1 has two antennas, and its client C1 has one antenna. AP2 has three antennas, and its clients C2 and C3 both have two antennas, referring to Figure 7. Suppose C1 is transmitting to AP1. In today’s network, AP2 is not allowed to transmit since it will cause interference at AP1. Using 802.11n+, AP2 can send signals \(x_2\) and \(x_3\) to its clients C2 and C3, respectively. To accomplish that, at AP1, AP2 aligns its two received signals from \(x_2\) and \(x_3\) in the direction orthogonal to the received signal from \(x_1\), so AP1 can recover \(x_1\) through orthogonal projection. At C2, AP2 aligns the received signal from \(x_3\) in the same direction with the ongoing received signal from \(x_1\). Thus through interference alignment, C2 can decode its desired signal \(x_2\). Similarly, by using interference alignment, C3 can decode its desired signal \(x_3\).

B. The 802.11n+ Channel Access Protocol

The 802.11n+ channel access protocol allows nodes with any number of antennas to contend for both time and degrees of freedom. Also, it provides bit rate selection.

In 802.11n+, a light-weight RTS-CTS is used [17], which is based on the observation that channel conditions stay the same in a duration of several milliseconds [28]. Specifically, instead of using regular RTS-CTS, the headers of the data packet and ACK are split out and transmitted first. The headers of the data packet and ACK are used in effect as RTS and CTS, respectively. This reduces the overhead significantly.

With 802.11n+, nodes listen to the medium using carrier sense. If the channel is idle, nodes contend for the medium using 802.11’s contention window and exponential backoff [1]. The node pair that won the contention exchanges a light-weight RTS-CTS. The RTS-CTS enable the nodes that intend to contend for the remaining degrees of freedom to estimate the channels to the receivers that won earlier contentions. The RTS-CTS also includes the number of antennas that will be used in the transmissions. After the RTS-CTS, the node pair exchanges the data packet, followed by the ACK.

In 802.11n+, nodes with more antennas than the number of used degrees of freedom contend for concurrent transmission. The number of used degrees of freedom is the number of ongoing transmissions, which the nodes can learn form the prior RTS-CTS messages. When a node performs carrier sensing, it ignores the signals from ongoing transmissions by projecting onto a subspace orthogonal to the ongoing transmissions.

With 802.11n+, a node that joins ongoing transmissions ends its transmissions at the same time as the ongoing transmissions, which the node learns from the prior RTS-CTS exchanges. This makes the medium idle at the end of each joint transmissions and provides starving nodes a fair chance.
to contend. After the joint transmissions of data packets, the ACKs are transmitted jointly as well using the same interference nulling and alignment scheme.

C. The 802.11n+ Concurrent Transmission Protocol

The concurrent transmission protocol described below applies to the transmissions of RTS, CTS, data, and ACK packets.

1) Notations: Let \( K \) denote the number of ongoing transmissions on the medium, \( TX \) denote an \( M \)-antenna transmitter that intends to join the ongoing transmissions, and \( m \) denote the maximum number of signals that \( TX \) can transmit without disrupting ongoing transmissions. For each signal \( x_i \), \( TX \) transmits \( V_i x_i \), where \( V_i \) is an \( M \)-dimensional precoding vector. Let \( R \) denote the set of receivers of the ongoing transmissions, and \( R' \) denote the set of receivers of \( TX \). An \( N \)-antenna receiver \( RX \) has a set of \( n \leq N \) desired signals called wanted signals, with the rest of the signals being unwanted ones. Let \( U \) denote the unwanted space and \( U^c \) the subspace orthogonal to \( U \).

2) The Transmission Protocol: The aim of the transmission protocol is to compute the precoding vectors such that \( TX \) can deliver its signals without interfering with the ongoing transmissions. The protocol is carried in three steps.

• Decide whether to null or align: If the receiver has an unwanted space, i.e., \( N > n \), then \( TX \) aligns the new interferences in the unwanted space, otherwise, i.e., \( N = n \), \( TX \) nulls all the new interferences.

• Compute the maximum number of signals to transmit: The \( M \)-antenna \( TX \) can transit \( m = M - K \) signals without disrupting the \( K \) ongoing transmissions, whose proof is based on the following two claims.

Claim 1 (nulling constraint): A transmitter can null its signals at a receiver with \( N \) antennas and \( n = N \) wanted signals by satisfying

\[
H_{N \times M} V_i = 0_{n \times 1}, \quad \forall i = 1, \ldots, m
\]  

where \( H_{N \times M} \) is the channel matrix, and \( 0_{n \times 1} \) is a \( n \)-dimensional column vector of all zeros.

Claim 2 (alignment constraint): A transmitter can align its signals in the unwanted space \( U \) of a receiver with \( N \) antennas and \( n < N \) wanted signals by satisfying

\[
U''_{n \times N} H_{N \times M} V_i = 0_{n \times 1}, \quad \forall i = 1, \ldots, m.
\]  

Based on the above two claims, a receiver \( i \) with \( n_i \) received signals corresponds to a matrix equation with \( n_i \) rows. The total number of constraint equations is equal to the total number of rows \( \sum n_i = K \). Since it has \( M \) antennas, \( TX \) can deliver \( m = M - K \) additional signals.

• Compute the precoding vectors: Using (15) and (16), we can compute the precoding vectors.

The transmission bit rate is selected on a per-packet basis as follows. The receiver uses the RTS to estimate the effective SNR (ESNR) after projection onto the subspace orthogonal to the ongoing transmissions. The receiver uses the ESNR to obtain the bit rate using the table given in [12], and sends the bit rate to the transmitter in CTS.

The testbed evaluations performed in [16] show that even for a small network with three contending node pairs, the average network throughput doubles that of a traditional network.

IV. JOINT MULTI-USER BEAMFORMING

In [22], a netMIMO transmission scheme called joint multi-user beamforming (JMB) is proposed, which is applicable to WLAN, especially in dense deployments such as enterprises, hotels, conference rooms, etc. The key contribution of JMB is a low-overhead technique for synchronizing the phases of multiple transmitters in a distributed manner. JMB allows a WLAN to scale up by adding more APs on the same channel. A similar scheme is proposed in [3].

The main idea of JMB is to elect one AP as the master and use its phase as a reference for all other APs (slaves). In JMB, before transmitting a data packet, the master transmits a few symbols. Slaves use those symbols to correct their phase errors. In JMB, APs are connected by a high-speed Ethernet. The packets intended for clients are distributed to all APs. The APs transmit concurrently to multiple clients, potentially delivering as many streams as the total number of antennas in all the APs. In the following, we first describe how the phases are synchronized, and then how the JMB transmissions are performed.

A. Distributed Phase Synchronization

To simplify the description, we consider a scenario of 2 single-antenna APs transmitting to 2 single-antenna clients. Let \( h_{ij}, i, j \in \{1, 2\} \) denote the channel coefficient from AP \( j \) to client \( i \). Let \( x_i(t) \) denote the symbol to be delivered to client \( i \) at time \( t \). Let \( y_i(t) \) denote the symbol received by client \( i \) at time \( t \). Let \( H = [h_{ij}] \) denote the 2×2 channel matrix, \( x(t) = [x_1(t), x_2(t)]^T \) denote the symbol vector to be delivered, and \( y(t) = [y_1(t), y_2(t)]^T \) denote the received symbol vector.

The main challenge for JMB is that the transmitters have different oscillators, which causes differences in their carrier frequencies, which in turn causes their phases to diverge. In the following, we first describe the case without oscillator offsets between the APs and the clients, and then the case with oscillator offsets.

1) Without Oscillator Offset: We use zero-forcing beamforming (ZFBF) for transmission [20], i.e., the transmitted signal \( s(t) = [s_1(t), s_2(t)]^T \) is given by

\[
s(t) = H^{-1}x.
\]  

The received signal is given by

\[
y(t) = H H^{-1} x(t) = x(t).
\]  

Thus, the effective channel is an identity matrix and each client can receive its symbol without interference from the other client’s signal.
2) With Oscillator Offset: Let $w_j^T$ denote the oscillator frequency for AP $j$, $w_i^R$ denote the oscillator frequency for client $i$. The channel matrix is given by $H(t) = [h_{i,j} e^{j(w_j^T - w_i^R)t}]$. Since the oscillators rotate with respect to each other, the channel does not have a fixed phase. We can decompose the channel matrix as $H(t) = R(t)HT(t)$, where $H = [h_{i,j}]$ is time invariant, and the diagonal matrices $R(t)$ and $T(t)$ are defined as

$$
\begin{pmatrix}
e^{-jw_j^Tt} & 0 \\
0 & e^{-jw_i^Rt}
\end{pmatrix}
$$

and

$$
\begin{pmatrix}
e^{jw_j^Tt} & 0 \\
0 & e^{jw_i^Rt}
\end{pmatrix}
$$

Let the transmitted signal be $s(t) = T(t)^{-1}H^{-1}x(t)$, then the received signal is given by

$$
y(t) = R(t)HT(t)T(t)^{-1}H^{-1}x(t) = R(t)x(t). \quad (19)
$$

The above equation is unchanged when we multiply by $1 = e^{jw_j^T} e^{-jw_i^R}$, i.e.,

$$
H(t) = e^{jw_j^T} R(t)HT(t)e^{-jw_i^R} = \begin{pmatrix}
e^{j(w_j^T - w_i^R)t} & 0 \\
0 & e^{j(w_j^T - w_i^R)t}
\end{pmatrix} H \begin{pmatrix}
1 & 0 \\
0 & e^{j(w_i^R - w_i^R)t}
\end{pmatrix}. \quad (20)
$$

The new effective channel is still diagonal. One AP serves as master, whereas other APs serve as slaves. The slaves measure the phase offsets with the master and correct them appropriately. As a result, the behavior of the APs emulates a single MIMO transmitter.

B. JMB Transmission Protocol

The JMB netMIMO transmission protocol works in two phases: channel measurement and data transmission, which are described below.

- **Channel Measurement:** In this phase, the APs measure two types of channels: 1) the channels from themselves to the clients ($H$), and 2) the channels from the master to the slaves, which provides the phase offset between the master and the slaves.
- **Data Transmission:** In this phase, the APs transmit jointly to deliver packets to multiple clients by using beamforming after correcting phase offsets with respect to the master. The symbol level synchronization in [21] is used. The APs start in the channel measurement phase, which is followed by multiple data transmissions. We only need to measure channel in an interval on the order of channel coherence time.

JMB is implemented using the off-the-shelf IEEE 802.11n cards in a testbed of 10 APs [22]. Results show that the throughput scales linearly with the number of APs, with a median gain of 8.1x to 9.4x.

V. NEMOX

Currently, netMIMO suffers from low throughput efficiency, defined as the ratio of network throughput to physical layer bit rate. The throughput efficiency in 300 Mbps 802.11n MIMO network is around 20%, and that in 802.11ac MU-MIMO network is below 10% [27]. One of the main reasons is that the multiplexing gain does not scale well in multi-cell WLANs. Deploying more APs does not solve the problem, because the inter-cell interference may eliminate the intra-cell multiplexing gain.

In [29], a scheme called NEMOx is proposed to enable scalable netMIMO as AP density and network size increase. With NEMOx, the network is organized into practical-sized clusters. A cluster is composed of a master AP (mAP) and a number of distributed APs (dAPs). With the coordination of the mAP, the APs within each cluster opportunistically synchronize with each other to perform netMIMO. A decentralized algorithm is used to manage inter-cluster interference and the tradeoff between spatial reuse and the AP’s cooperation gain.

The deficiency with previous netMIMO schemes can be demonstrated by the following example. Suppose we have two dAPs, dAP1 and dAP2, and two clients, client1 and client2. Both clients are close to dAP1 but far away from dAP2. Both dAPs have the same power budget. Thus, dAP2 uses full power and dAP1 must reduce its power so that the crosstalk interference can be canceled. The reduced power leads to lower data rates for both clients. In such case, higher data rate may be obtained by serving a single client with full power. The insight obtained from this example is that optimizing netMIMO involves not only ZFBF precoding but also the allocation of power of dAPs and the selection of clients to be served.

In the following, we first describe NEMOx’s channel access protocol and then describe its power budgeting and client selection methods.

A. Channel Access

Similar to previous netMIMO schemes [4], [22], NEMOx focuses on downlink transmissions from APs to clients. In a NEMOx cluster, the dAPs function as a set of distributed antennas, which performs carrier sensing and transmits modulated signals. The mAP manages dAP’s PHY-layer modulation/demodulation and MAC-layer channel access. The PHY provides carrier sensing information to the MAC. The MAC manages the channel access opportunities, informing the PHY which dAPs to use and when to start transmission. In the following, we first describe the mechanism for creating virtual APs and that for medium access control, then we describe the optimizations to improve performance.

1) Creating Virtual APs: When a subset of dAP all sense an idle channel, the mAP may choose to select them into a virtual AP (vAP) and perform a probabilistic contention algorithm to acquire the channel and avoid collision with dAPs in other clusters. The vAP keeps a single back-off timer, whose expiration triggers the netMIMO transmission of the vAP. All dAPs in the vAP are synchronized, whereas different vAPs contend with each other asynchronously.


**Dominance Relationship:** We establish a dominance relationship among vAPs as follows. A vAP $V_i$ dominates another vAP $V_j$ within the same cluster if $V_j$’s dAPs is a subset of $V_i$’s, and $V_j$’s interfering dAPs in the neighboring clusters are a subset of $V_i$’s. Thus, $V_i$ has a higher multiplexing gain and a lower number of contenders, and thus is more preferable than $V_j$. The dominated vAPs are pruned by the mAP in each cluster to simplify the management. To establish a dominance relationship, the vAPs sample the channel status. If vAP $V_j$ is always busy whenever $V_i$ is busy, but not vice versa, then $V_i$’s interfering dAPs are a subset of $V_j$’s. The mAP coordinates such sampling and refreshes the dominance relationship periodically.

2) **Semi-synchronized CSMA:** In the following, we describe the channel access protocol of NEMOx, which consists of client association and state transition.

**Client Association:** When a client has packets to receive, all dAPs that can deliver the packets are potential transmitters. All vAPs that contain such potential transmitters will trigger channel contention. A client may be served by a different vAP for each transmission attempt. Such a soft association with the client makes it possible to vary the throughput of a client by matching it with vAPs of different sizes.

**State Transition:** The time is divided into slots, with slot duration the same as that of IEEE 802.11, i.e., 9 $\mu$s. Slots are synchronized only in vAPs in the same cluster. NEMOx runs the following probabilistic channel-access algorithm.

- A vAP $V_j$ starts in the busy state by default. It enters the idle state when all of its dAPs sense an idle channel for a fixed number of slots, similar to DIFS defined in IEEE 802.11.
- Then, $V_j$ enters the contention state with a contention probability $p_j$. $V_j$ remains in the idle state with probability $1 - p_j$ and increases its contention probability as $p_j = p_j + \delta_1$, where $\delta_1 \in (0, 1)$ is a parameter that indicates the aggressiveness $V_j$ increases its contention probability.
- After entering the contention state, $V_j$ starts the back-off and selects a random number between $0$ and $B$, where $B$ is the back-off window size. When the back-off expires, $V_j$ starts netMIMO transmission.
- If $V_j$ senses a busy channel during the back-off, it declares a contention failure and enters the busy state. $V_j$ then reduces its contention probability as $p_j = p_j - \delta_2$, where $\delta_2 \in (0, 1)$ is a parameter to be optimized later.
- If the transmission is successful and $V_j$ receives ACKs from all clients, $V_j$ returns to the initial busy state and updates its contention probability as $p_j = p_j + \delta_1$. Otherwise, a transmission failure is declared and $V_j$ updates its contention probability as $p_j = p_j - \delta_2$.

Collision can occur only among vAPs of different clusters in NEMOx. When vAPs in a cluster start back-off counters at the same time, the mAP randomly enables one of them. In NEMOx, medium access control occurs at the group (vAP) level rather than at the node (dAP) level, leading to much less number of contention domains than that of traditional IEEE 802.11 WLANs.

3) **Optimizing Media Access:** In the following, we first formulate the optimization problem, then describe a decentralized solution and its implementation in NEMOx.

**Formulating the Optimization Problem:** The goal of the MAC is to provide optimal and fair channel access for all dAPs. Let $r_i$ denote the probability of successful transmission for dAP $i$, and $C$ the set of clusters. The optimization problem can be formulated as

$$
\max \sum_{K \in C} \sum_{i \in K} \alpha_K U(r_i)
$$

where $K$ is the cluster that dAP $i$ is in, $\alpha_K$ is the weight of $K$ used for prioritization, and $U(r_i)$ is the utility function for $r_i$. When $U(r_i) = \log(r_i)$, maximizing the sum utility achieves both global optimality with respect to $r_i$ and proportional fairness [13]. The above optimization problem is subject to the following two constraints.

The first constraint is that CSMA allows at most one vAP to transmit in a contention domain. This constraint can be modeled as a maximal clique, which is a complete graph where there is edge between two vAPs if they interfere with each other. The sum contention probability of all vAPs in the maximal clique should not exceed 1. Let $M$ denote the set of maximal cliques in the network, then we have

$$
\sum_{j \in m} p_j \leq 1, \forall m \in M.
$$

The second constraint is as follows. The effective transmission probability $r_i$ of dAP $i$ is the sum transmission probabilities of all vAPs containing dAP $i$. Let $f_j$ denote the probability of VAP $j$ suffering a transmission/contention failure. The effective transmission probability of VAP $j$ is $(1 - f_j)p_j$. Thus, we have

$$
r_i = \sum_{j \in e_j} (1 - f_j)p_j.
$$

Since the utility function is defined with respect to each dAP, the solution to the optimization problem in [21] favors vAPs with more dAPs, leading to a higher multiplexing gain. The solution to [21] provides the channel contention probability $p_j$ for each vAP $j$. However, solving the optimization problem is complex, involving global exchange of information among all vAPs and dAPs. In the following, we describe a decentralized optimization solution that can still achieve the optimization objective.

**Decentralized Optimization Solution:** We obtain the decentralized solution by decomposing the variables into subsets, each associated with one cluster. Since [21] is a convex optimization problem, we can obtain its Lagrangian as

$$
L = \sum_{K \in C} \sum_{i \in K} \alpha_K U(r_i) - \sum_{m \in M} \sum_{j \in m} \beta_m p_j
$$

where $\beta_m$ is the Lagrange multiplier for the maximal clique $m$. We have $\beta_m = \beta f_m$, where $f_m$ is the collision probability in maximal clique $m$. Since vAP $j$ may belong to multiple maximal cliques, its collision probability is given by

$$
f_j = 1 - \prod_{m \in m} (1 - f_m) \approx \sum_{m \in m} f_m.
$$
The above approximation is valid when $f_m$ is kept small. This can be achieved by keeping a reasonably small value for $\delta_1$ and a large value for $B$, where $\delta_1$ and $B$ are channel access parameters described in the previous subsection, so that no vAP becomes overly aggressive in contending the channel. A similar assumption has been made in traditional utility-based wireless MAC design [19].

Using the approximation in (25), $f_j$ can be obtained by making vAP $j$ to locally track its own contention/failure probabilities, without the knowledge of its maximal cliques or loss probabilities in each of the cliques. Substituting for $\beta_m$ and using (25), we have $L = \sum_{K \in C} L_K$ and $L_K$ is given by

$$L_K = \sum_{i \in K} \alpha_i U(r_i) - \beta \sum_{i \in K, j \in E} f_j p_j.$$  \hfill (26)

It has been shown that the aggregate utility for a system of equations of the above form is maximized when the individual utilities are maximized [13], [19]. The individual utility corresponds to a cluster in $K$. This requires joint optimization of vAPs within a cluster. Apply the KKT conditions to each vAP $j$, we have

$$\frac{dL_K}{dp_j} = \sum_{i \in j} \alpha_i U(r_i)(1 - f_j) - \beta f_j = 0.$$  \hfill (27)

Since $U(r_i) = \log(r_i)$, we have

$$\frac{dL_K}{dp_j} = \sum_{i \in j} \alpha_i (1 - f_j) r_i^{-1} - \beta f_j = 0$$  \hfill (28)

which can be rewritten as

$$\alpha_i (1 - f_j) - \beta f_j (\sum_{i \in j} r_i^{-1})^{-1} = 0.$$  \hfill (29)

We can use $\alpha_i$ to prioritize different clusters, but here we set a constant $\alpha_i = \alpha$. Since $f_j$ can be obtained locally, the above optimality condition can be achieved by adapting each vAP $j$’s contention probability in a decentralized manner as

$$p_j \leftarrow p_j + \alpha - f_j (\beta (\sum_{i \in j} r_i^{-1})^{-1} + \alpha).$$  \hfill (30)

Based on the KKT condition, the adaptation mechanism for each vAP uses a gradient approach. Since both the utility function and the Lagrangian are concave with respect to $p_j$, there is a unique maximum, to which the adaptation converges [19].

**Implementing Optimization Solution in NEMOx:** In NEMOx, the adaptive mechanism occurs in discrete time frames, each in one of the three states: idle, transmission success, and transmission/contention failure. At the end of the frame, a vAP $j$ updates its contention probability, i.e., $\delta_1 = \alpha$. There is a tradeoff in selecting $\alpha$. Larger $\alpha$ values make the adaptation faster, but can cause oscillation around the optimal value. Initially, we can set $\alpha = 0.05$.

Using (30), vAP $j$ reduces its contention probability $p_j$ by $\delta_2 = p_j (\beta (\sum_{i \in j} r_i^{-1})^{-1} + \alpha)$ whenever a transmission/contention failure occurs. Larger $\beta$ values incur higher penalty to $p_j$ when collision occurs, but can cause oscillation around the optimal value. Here we can set $\beta = 0.25$.

When interfering vAPs increase their contention probability to high values, severe collision may occur. To solve this problem, NEMOx uses an additional back-off and randomizes the transmissions within the back-off window of size $B$. The value of $B$ need not to be adaptive, since the contention probability is already adaptive. Here we can set $B = 32$.

**B. Power Budgeting and Client Selection**

In the following we describe how to optimize power budgeting and client selection. The assumption in this section is that a vAP has won channel contention.

1) Power Budgeting: NEMOx seeks to optimize the multiplexing gain and the diversity gain using the following joint precoding and power (JPP) optimization formulation

$$\max \sum_{i=1}^{|D|} w_i \log_2(1 + \frac{P_i}{N_0})$$

subject to

$$P_i = \sum_{k=1}^{|S|} h_{i,k} v_{k,i}^2, \forall i \in D$$

$$\sum_{i=1}^{|D|} |v_{k,i}|^2 \leq P_{max}, \forall k \in S$$

$$\sum_{k=1}^{|S|} h_{j,k} v_{k,i} = 0, \forall i \in D, j \neq i$$  \hfill (31)

where $D$ and $S$ are the set of clients and dAPs, respectively. The objective is to maximize the weighted sum data rate of the clients, where $w_i$ is the weight that is adjusted for fairness and will be discussed later, and Shannon’s channel capacity formula is used. The second line of the optimization formulation computes the received power for client $i$ with precoding $v_{k,i}$ and channel response $h_{k,i}$ with respect to dAP $k$. The third line is the power budget constraint for each dAP. The last line is the precoding constraint, which says that the precoded data symbols from $i \neq j$ must cancel each other out when arriving at client $j$ through the wireless channels. The output of JPP is the precoding $v_{k,i}$ and the power budget $P_i$.

When the topology is imbalanced, JPP favors a few clients with high capacity rather than many clients with low capacity. Such tradeoff between multiplexing gain and diversity gain has not been addressed previously.

The JPP formulation is non-convex with respect to the real and imaginary components of $v_{k,i}$ because of the norm operation in the second line of (31). However, we can impose $\text{Im}(\sum_{k=1}^{|S|} h_{i,k} v_{i,k}) = 0$, by phase-shifting the vector $v_{i,k}, \forall k \in S$, appropriately. The third and fourth lines of (31) are invariant under phase-shift. The resulting problem is convex and can be efficiently solved by standard convex optimization techniques.

2) Client Selection: NEMOx uses an opportunistic client selection algorithms that incorporate JPP precoding to achieve fairness, see Algorithm [1].

The algorithm iterates in rounds. In each round, we search for the bestClient that maximizes the weighted sum rate when it joins with those clients already selected. The selection stops when the number of selected clients exceeds that of dAPs, or...
Algorithm 1 Opportunistic Client Selection

Input: $S$: a set of dAPs in a vAP that has won contention; $D$: a set of clients that can be served by the dAPs in $S$; $E$: a set of clients in this cluster.
1: Initialization: the set of currently selected dAPs $D' = \emptyset$, the current optimal sum capacity $O_{max} = 0$
2: while $|D'| < |S|$ do
3: for all $i \in D \setminus D'$ do
4: Solve JPP, with the set of dAP $S$ and the set of clients $i \cup D'$, obtain the optimal sum capacity $O_i$
5: if $O_i \geq O_{max}$ then $O_{max} = O_i$, bestClient = $i$
6: end if
7: end for
8: $D' = \text{bestClient} \cup D'$
9: end while
10: Run netMEMO transmission from $S$ to $D'$
11: Client $i$’s throughput is $R_i$
12: $R_i = \gamma R_i + (1 - \gamma) R_{\text{avg}}$
13: $w_i = 1/R_i$ for all $i \in D'$. The selection of the bestClient yields a lower sum rate than that of the previous iteration. Then, we perform netMIMO transmission with client $i$’s throughput being $R_i$. We update the time-averaged throughput $R_{\text{avg}}$ using a moving average with a smoothing factor $\gamma$, which is set to 0.1. Finally, we make the throughput weight $w_i$ be inverse of $R_i$ to achieve long-term proportional fairness [28].

3) NEMO’s netMIMO Transmission: NEMO’s netMIMO transmission is performed as follows. 1) The dAPs broadcast the same RTS packet synchronously, which includes the transmission duration and clients’ addresses and synchronizes the clients. 2) All the selected clients send back the same CTS packet synchronously, which includes the transmission duration and reserves the channel from transmitters in neighboring clusters. 3) The clients sequentially send out the channel state information (CSI), with the channel vector being from the dAPs to the client. CSI is used for computing the precoding.

We abort the netMIMO transmission if CSI from any client is missing, which can be caused by CTS channel reservation failure. In such case, other vAPs can reuse the channel using NEMOx’s channel access protocol. 4) If the netMIMO transmission succeeds, the clients sequentially send an ACK.

NEMOx is evaluated using a software-radio testbed in [29]. It is shown that NEMOx has throughput scalability and has multiple-fold performance gain over existing wireless LAN architecture and other netMEMO schemes.

VI. CONCLUSION

In this paper, we describe some representative netMIMO methods. We show that netMIMO can significantly increase the network capacity. We believe the netMIMO methods described here are just the beginning of the new technologies to address the challenge of ever-increasing wireless traffic demand, and the future will see further exciting developments in this field.

REFERENCES

[1] IEEE Std 802.11-1997, pp. i-445, 1997.
[2] System Description and Operating Principles for High Throughput Enhancements to 802.11, IEEE 802.11-14/0870r, 2004.
[3] E. Arayafar, N. Anand, T. Salonidis, and E. W. Knightly, “Design and experimental evaluation of multi-user beamforming in wireless LANs,” in Proc. of ACM MobiCom, pp. 197-208, 2010.
[4] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, “Achieving high data rates in a distributed MIMO system,” in Proc. of ACM MobiCom, 2012.
[5] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, “AirSync: enabling distributed multiuser MIMO with full spatial multiplexing,” IEEE Transactions on Networking, Vol. 21, No. 6, 2013.
[6] Cisco Global Mobile Data Traffic Forecast Update: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827_white_paper_c11-520862.html
[7] D. Gesbert, M. Kountouris, R. W Heath, C.-B. Chae, and T. Salzer, “Shifting the MIMO paradigm,” IEEE Signal Processing Magazine, Vol. 24, No. 5 pp. 36-46, 2007.
[8] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, “Multi-cell MIMO cooperative networks: a new look at interference,” IEEE Journal on Selected Areas in Communications (JSAC), Vol. 28, No. 8, pp. 1300-1408, 2010.
[9] S. Gollakota, S. D. Perli, and D. Katabi, “Interference alignment and cancellation,” in Proc. of ACM SIGCOMM, 2009.
[10] S. Gollakota, S. D. Perli, and D. Katabi, “Overcoming the antennas-per-node throughput limit in mimo lans,” Technical report, MIT, 2009.
[11] M. Guillaud, D. Slock, and R. Knopp, “A practical method for wireless channel reciprocity exploitation through relative calibration,” in Signal Processing and Its Applications, pp. 403-406, 2005.
[12] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Predictable 802.11 packet delivery from wireless channel measurements,” in Proc. of ACM SIGCOMM, 2010.
[13] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” The Journal of the Operational Research Society, Vol. 49, No. 3, pp. 237252, 1998.
[14] G. Kramer, I. Maric, and R. D. Yates, “Cooperative communications,” Foundations and Trends in Networking, Vol. 1, No. 3, pp. 271425, 2006.
[15] J. W. Lee, M. Chiang, and R. A. Calderbank, “Optimal MAC design based on utility maximization: Reverse and forward engineering,” in Proc. IEEE INFOCOM, 2006.
[16] K. C.-J. Lin, S. Gollakota, and D. Katabi, “Random access heterogeneous MIMO networks,” in Proc. of ACM SIGCOMM, 2011.
[17] K. C.-J. Lin, Y. Chuang, and D. Katabi, “A light-weight wireless handshaking device using NEMOx’s channel access protocol,” in Proc. of ACM SIGCOMM, 2010.
[18] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Std. 802.11ac Draft 3.0, 2012.
[19] T. Nandagopal, T. E. Kim, X. Gao, and V. Bhargavan, “Achieving MAC layer fairness in wireless packet networks,” in Proc. of ACM MobiCom, 2000.
[20] B. C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, “A vector-perturbation technique for near-capacity multiantenna multiuser communication - Part I: channel inversion and regularization,” IEEE Transactions on Communications, Vol. 53, No. 1, pp. 195-202, 2005.
[21] H. Rahul, H. Hassanieh, and D. Katabi, “Sourcesync: a distributed wireless architecture for exploiting sender diversity,” in Proc. of ACM SIGCOMM, 2010.
[22] H. S. Rahul, S. Kumar, and D. Katabi, “JMB: Scaling wireless capacity with user demands,” in Proc. of ACM SIGCOMM, 2012.
[23] S. Saunders, S. Carlaw, A. Giustina, R. R. Bhat, S. Rao, and R. Siegel, Femtocells: Opportunities and Challenges for Business and Technology, Wiley, 2009.
[24] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno, “Coordinated multipoint transmission/reception techniques for LTE-advanced Coordinated and Distributed MIMO,” IEEE Wireless Communications, Vol. 17, No. 3, pp. 26-34, 2010.
[25] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, “Argos: Practical many-antenna base stations,” in Proc. ACM MobiCom, Istanbul, Turkey, 2012, pp. 5364.
[26] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G.M. Voelker,”SAM: Enabling practical spatial multiple access in wireless LAN,” in Proc. of ACM MobiCom, pp. 49-60, 2009.
[27] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, “Fine-grained channel access in wireless LAN,” in Proc. of ACM SIGCOMM, 2010, pp. 147-158.
[28] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[29] X. Zhang, K. Sundaesan, M. A. Khojastepour, S. Rangarajan, and K. G. Shin, "NEMOx: scalable network MIMO for wireless networks," in *Proc. of ACM MobiCom*, 2013, pp. 453-464.