# NEMOx: Scalable Network MIMO for Wireless Networks

Xinyu Zhang
University of Wisconsin-Madison
xyzhang@ece.wisc.edu

Karthikeyan Sundaresan
Mohammad A. (Amir) Khojastepour
Sampath Rangarajan
NEC Laboratories America
{karthiks,amir,sampath}@nec-labs.com

Kang G. Shin
University of Michigan-Ann Arbor
kgshin@eecs.umich.edu

## ABSTRACT

Network MIMO (netMIMO) has potential for significantly enhancing the capacity of wireless networks with tight coordination of access points (APs) to serve multiple users concurrently. Existing schemes realize netMIMO by integrating distributed APs into one "giant" MIMO but do not scale well owing to their global synchronization requirement and overhead in sharing data between APs. To remedy this limitation, we propose a novel system, NEMOx, that realizes netMIMO downlink transmission for *large-scale* wireless networks. NEMOx organizes a network into practical-size clusters, each containing multiple distributed APs (dAPs) that opportunistically synchronize with each other for netMIMO downlink transmission. Inter-cluster interference is managed with a decentralized channel-access algorithm, which is designed to balance between the dAPs' cooperation gain and spatial reuse—a unique tradeoff in netMIMO. Within each cluster, NEMOx optimizes the power budgeting among dAPs and the set of users to serve, ensuring fairness and effective cancellation of cross-talk interference. We have implemented and evaluated a prototype of NEMOx in a software radio testbed, demonstrating its throughput scalability and multiple folds of performance gain over current wireless LAN architecture and alternative netMIMO schemes.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless Communications*; C.2.2 [**Computer-Communication Networks**]: Network Protocols

## General Terms

Algorithms, Design, Experimentation, Performance

## Keywords

Network MIMO, multi-user MIMO networks

## 1. INTRODUCTION

Despite the well-known bit-rate enhancements via multiplexing gain, the throughput of MIMO networks has been improving at a much slower pace. *Throughput-efficiency*—defined as the ratio of network throughput to PHY bit-rate—is only around 20% in 300Mbps 802.11n MIMO networks, and below 10% in Gbps 802.11ac MU-MIMO networks [1]. One major reason for this is that the multiplexing gain does not scale to multi-cell WLANs that involve spatially-distributed transmitters. Deploying more APs does not solve this problem either, as inter-cell contention overhead may undo the intra-cell multiplexing gain from MIMO.

Network MIMO (netMIMO) [2], also referred to as *distributed MU-MIMO*, has potential for eliminating inter-cell interference, and hence eliminating contention, by allowing APs to tightly synchronize and share the transmission of data packets. An ideal netMIMO system can enable concurrent transmissions from distributed APs, thus scaling the downlink capacity linearly with the number of APs in a network. Researchers have explored the design of netMIMO by combining distributed APs into one giant MIMO transmitter [3–6]. APs are completely synchronized at the carrier-signal level by using a reference signal, and share each other's data (for cancelling cross-talk interference) via a wireline backhaul. While these schemes are promising for small-scale networks within a single contention domain, their tight synchronization requirement renders them infeasible for large-scale multi-cell wireless networks. The need for sharing data packets and channel state in real time may also deplete the backhaul capacity even if an expensive fiber backhaul connection were used [2].

We believe that the eventual means of realizing netMIMO is a hierarchical architecture that decomposes a large network into small, practical-size clusters, each containing multiple tightly coordinated APs running netMIMO. Distant clusters do not interfere with each other and can leverage spatial reuse; synchronization and data sharing are limited to within each cluster, thus eliminating the need for network-wide coordination. Realizing this vision, however, is very challenging. Interference between clusters, unless properly managed, can easily nullify the intra-cluster netMIMO gain. An extension of existing interference-avoidance protocols, which regards a cluster as one node, is feasible but insufficient—interference patterns can vary significantly over distributed APs even within one cluster, and thus coarse-grained interference avoidance may severely lower the spatial reuse.

In this paper, we design and implement a novel framework, called *NEMOx*, that realizes the potential of netMIMO for large-scale wireless networks. In NEMOx, a *cluster* contains one master AP (mAP) which coordinates a set of distributed APs (dAPs) (Figure 1) for netMIMO downlink transmission. The dAPs employ a distributed CSMA mechanism to avoid inter-cluster interference, which evades global synchronization and enables spatial reuse across clusters.

The nature of operation in netMIMO, however, poses a fundamental challenge. CSMA realizes asynchronous contention, but cannot handle the synchronous cooperation (*i.e.*, netMIMO trans-


*MobiCom'13,* September 30-October 4, Miami, FL, USA.

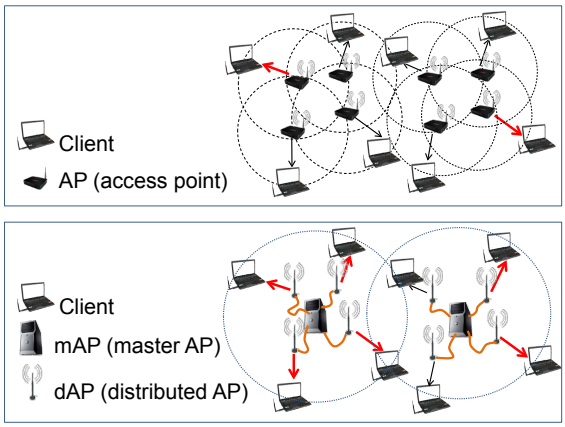


**Figure 1: Provisioning capacity and coverage using a traditional multi-cell WLAN (top) and NEMOx (bottom), respectively. Thick arrows indicate concurrent interference-free transmissions.**

mission) between distributed APs. NEMOx overcomes this difficulty via a semi-synchronized group-based contention mechanism, which prescribes subsets of dAPs as contention entities. At runtime, the mAP opportunistically schedules the subset with idle dAPs. Such a contention mechanism—contention being group-specific yet transmission being dAP-specific—is unique to netMIMO and has not been addressed in existing MAC designs. NEMOx proposes a simple, decentralized algorithm to optimize the contention between groups, and achieve proportionally fair channel access among dAPs.

While intra-cluster netMIMO transmission can be devised by migrating the 802.11ac MU-MIMO modulation algorithm that exploits multiplexing gain [5, 7], we find this could severely degrade netMIMO's performance. In particular, netMIMO may encounter unbalanced topologies where clients are concentrated near a few dAPs (which is common in, for example, conference rooms). In such cases, greedy exploitation of dAPs' multiplexing gain may result in low throughput for all clients. Further, since a client's throughput is coupled with dAPs and other clients within the same netMIMO transmission group, NEMOx meets these challenges by jointly optimizing the power budget of dAPs and opportunistically serving subsets of clients, so as to create balanced topologies and provide clients throughput fairness.

**Implementation.** We have prototyped NEMOx's MAC/PHY components on the WARP software radio platform [8]. Each cluster is built on one WARP node serving as an mAP connecting to, and controlling 4 dAPs (remote antennas). Sharing a central processing unit at the mAP, the dAPs achieve carrier-level synchronization without using an external clock, and real-time data sharing without a backhaul network, thereby making it possible to realize netMIMO transmission within a cluster. Our extensive experiments demonstrate that NEMOx effectively addresses the PHY-level challenges unique to netMIMO, allowing downlink throughput to scale linearly with the number of dAPs in a cluster. Experiments in a network testbed with multiple clusters and trace-driven emulation further verify the effectiveness of NEMOx's MAC in coordinating netMIMO clusters in large networks, yielding a multi-fold gain over current wireless LAN architecture and existing netMIMO schemes.

NEMOx confines its main operations to the mAP and requires no modification to client receivers. Further, its approach of building netMIMO preserves the signal processing modules in 802.11ac hardware, thus facilitating its deployment by simply upgrading the mAP's MAC driver and distributing its transmission points (antennas) using RF cables. Note that NEMOx's MAC/PHY components are not tied to this deployment model. Existing approaches [5, 6] to building single-cluster netMIMO can also be used in our framework.

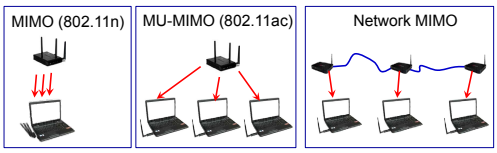


**Figure 2: Wireless LAN architecures other than SISO, and NEMOx shown in Figure 1.**

**Contributions.** We make several contributions via NEMOx as follows:

- Proposed a scalable, hierarchical architecture for netMIMO in wireless networks that realizes netMIMO efficiently within small clusters and employs asynchronous random access to enable spatial reuse across clusters.
- Designed a protocol to manage the synchronous cooperation and asynchronous channel access for dAPs in netMIMO. This protocol balances between multiplexing gain and spatial reuse, allowing network capacity to scale with the number of clusters.
- Optimized the netMIMO scheme that accounts for joint power control and client selection within each cluster, allowing a cluster's capacity to scale with the number of dAPs.
- Developed a simple way to build netMIMO clusters and implemented a working prototype of NEMOx in a WARP-based testbed, demonstrating NEMOx's effectiveness.

The rest of the paper is organized as follows. Sec. 2 briefly introduces MIMO-based network architectures. Sec. 3 presents an overview of the NEMOx architecture while Sec. 4 and 5 detail the design of NEMOx. Sec. 6 and 7 describe the implementation and experimental validation of NEMOx. Sec. 8 discusses NEMOx's limitations and our future work. Sec. 9 covers related work and finally, Sec. 10 concludes the paper.

## 2. BACKGROUND

Over the last several years, there has been an explosive growth of mobile traffic demand, whereas wireless spectrum remains a scarce resource, and this trend is expected to continue in future. It is, therefore, important to increase the transmission concurrency, or multiplexing gain. Ideally, a WLAN infrastructure should scale its multiplexing gain with the density of APs, while ensuring good coverage of a given area. Spatial reuse should also be exploited to scale the network capacity as its network area expands. Wireless LANs have evolved over the last decade with different architectures (Figure 2) to deal with these two levels of scalability.

Table 1 shows the capacity-scaling profiles of these WLAN architectures. For a fair comparison, they are assumed to use the same number of transmit antennas to cover the same area. Each transmit antenna has the same power budget, a practical constraint for radio hardware [2].

**SISO and MIMO.** In 802.11a/b/g SISO WLANs, neighboring APs interfere with each other and hence can only transmit alternately, *i.e.*, multiplexing gain is infeasible for transmitters in the same contention domain. However, APs in different contention domains can exploit spatial reuse to scale network capacity with network size. A MIMO AP, based on 802.11n, can send multiple streams of data to the *same client*, or beamform the same data through multiple antennas to improve SINR and link capacity, *i.e.*,

| | SISO | MIMO | CAS | netMIMO | NEMOx |
|---|---|---|---|---|---|
| Multiplexing gain | No | No | No | Yes | Yes |
| Diversity gain | No | Yes | Yes | Yes | Yes |
| Spatial reuse | Yes | Yes | Yes | No | Yes |
| Scalability, one contention domain | No | No | No | Yes | Yes |
| Scalability, multiple contention domains | Yes | Yes | Yes | No | Yes |

**Table 1: Scalability profiles for WLAN architectures. The multiplexing gain and diversity gain refer to the gains from cooperation across multiple WLAN cells.**

*diversity gain*. However, like SISO, neighboring MIMO APs are not cooperative, and hence provide no multiplexing gain among them.

**MU-MIMO or CAS.** The emerging 802.11ac multi-user MIMO (MU-MIMO) network [9] allows a multi-antenna AP to send different data streams concurrently to *multiple clients*. However, concurrency is achievable only for clients in the same cell, with all transmit antennas co-located at one AP. Like MIMO networks, such a co-located antenna system (CAS) cannot harness multiplexing gain for distributed APs, thereby not allowing the network capacity to scale with AP density.

**NetMIMO.** NetMIMO enables simultaneous transmission of different data streams through multiple dAPs that are tightly synchronized to form a "giant" multi-antenna transmitter [3–6]. The dAPs can serve their clients simultaneously by cancelling cross-talk interference, thus achieving multiplexing gain and scaling capacity with the AP density. However, existing netMIMO schemes require *full-synchronization* (*w.r.t.* transmission time, sampling clock-rate, carrier frequency and phase) and data sharing between all APs, which is not feasible for large networks with many APs spanning multiple contention domains.

NEMOx is designed as a practical solution to overcome this fundamental limitation, allowing netMIMO's downlink capacity to scale with both AP density and network size.

# 3. NEMOx: AN OVERVIEW

## 3.1 Motivation and Challenges

### 3.1.1 Why MAC for netMIMO?

A hierarchical architecture makes it feasible to deploy netMIMO for large-scale wireless networks. But a fundamental question associated with it is: *How should a cluster contend for channel access and avoid interference with its neighbors?*

Although the intra-cluster dAPs' operations can be centralized at the mAP, they may border on dAPs of different clusters, thus experiencing asynchronous channel states or interference patterns. Unfortunately, netMIMO transmission can be realized only when all dAPs in a cluster synchronize their transmission attempts, *i.e.*, each senses an idle channel and finishes a random backoff at the same time.

To overcome this dilemma, the mAP may opt to wait for all its dAPs to become available and then trigger synchronous netMIMO transmissions from them. While this approach would provide a maximum multiplexing gain, the strict binding of dAPs can severely reduce spatial reuse opportunity of those dAPs that have less contention with dAPs in other clusters. Alternatively, dAPs in each cluster can run CSMA independently and start synchronous netMIMO transmissions if they acquire channels simultaneously. However, such opportunities are rare due to the decentralized and asynchronous nature of CSMA. Thus, this approach significantly limits multiplexing gain and reduces NEMOx to an 802.11 multi-cell WLAN.

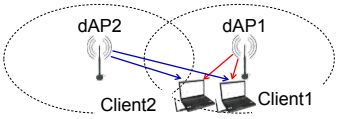


**Figure 3: In an unbalanced topology, concurrently serving two clients results in underutilization of the dAP's power budget, leading to low-rate for both.**

In arbitrating channel contention between clusters, the medium access mechanism in NEMOx must therefore *strike a balance between multiplexing gain and spatial reuse in coordinating dAPs within the same cluster*. Realizing this objective entails a new MAC design for netMIMO which (i) allows asynchronous contention across clusters to achieve scalability, (ii) employs flexible subsets of cooperating dAPs in each cluster to leverage spatial reuse, and (iii) opportunistically synchronizes dAPs within a cluster to realize netMIMO transmission and harvest multiplexing gain.

### 3.1.2 Tailoring netMIMO PHY for distributed APs

A common approach to realizing netMIMO transmission is to migrate the 802.11ac MU-MIMO PHY, treating dAPs as multiple antennas of a giant AP [5, 6], and using zero-forcing beamforming (ZFBF) to cancel cross-talk interference. ZFBF is a linear *precoding* mechanism that allows each dAP to send a weighted combination of the data symbols intended for different clients. Suppose $\boldsymbol{h}$ is the channel matrix and $\boldsymbol{x}$ the vector of symbols, then the weights, or so called the *precoding vector*, are created through a pseudo-inverse: $\boldsymbol{v} = \boldsymbol{h}'(\boldsymbol{h}\boldsymbol{h}')^{-1}$. Since $\boldsymbol{h}\boldsymbol{v}\boldsymbol{x} = \boldsymbol{I}\boldsymbol{x}$ where $\boldsymbol{I}$ is the identity matrix, each client only receives the data intended for itself. In other words, the clients' cross-talk interference is pre-cancelled by the dAPs.

However, we found ZFBF to be sub-optimal for netMIMO, especially in an unbalanced network topology. In MU-MIMO, transmit antennas are co-located and impose a similar level of cross-talk interference on each client, and hence, it is reasonable to use ZFBF which implicitly projects equal power for all clients [7]. But for netMIMO, due to the distributed nature of transmit antennas, the cross-talk interference may dominate the useful signals when clients are concentrated near one dAP.

For example, the two clients in Figure 3 are concentrated near dAP1. Since dAP2 is far away, it needs much greater power to cancel the cross-talk interference created by dAP1. However, since both dAPs have the same power budget, dAP2 should use its full power whereas dAP1 must reduce its power so that its cross-talk interference can still be cancelled by dAP2. As a result, only a small fraction of the power budget is used on useful signals, resulting in low rates for both clients. In such a case, it may be preferable to serve a single client (leveraging diversity gain) with full transmit power, or select clients that form a more balanced topology.

In addition, unlike in existing MAC/PHY protocols [10,11], where fairness can be defined *w.r.t.* a link (Tx–Rx pair), such a notion does not exist in netMIMO, where clients' rates are coupled and depend on a set of dAPs that are jointly serving them.

Therefore, *optimizing NEMOx's netMIMO operations involves not only ZFBF precoding, but also allocation of power from dAPs to clients, and the set of clients to be served on each transmission attempt.*

## 3.2 Basic Operations

NEMOx adopts a *thin AP* approach to building a netMIMO cluster. The dAPs in each cluster are simply a set of distributed anten-

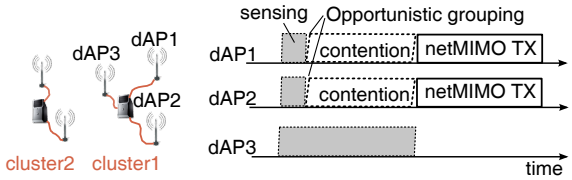


**Figure 4: Operations of an mAP controlling 3 dAPs. The mAP schedules dAPs' transmissions and modulates their data to remove cross-talk interference. The dAPs are responsible for carrier sensing and emitting modulated signals.**

nas responsible for carrier sensing and emitting modulated signals. Both their PHY-layer packet modulation/demodulation and MAC-layer channel access protocols are centrally managed by the mAP. The MAC/PHY follows a modular design with a simple interface. The PHY provides each dAP's carrier sensing decision to the MAC, whereas the MAC is responsible for acquiring channel access opportunities, telling the PHY when to start netMIMO transmission and which dAPs to employ. As in existing netMIMO schemes [5,6], NEMOx primarily focuses on concurrent *downlink transmissions* from dAPs to clients.

Figure 4 illustrates a typical operation flow of an mAP controlling 3 dAPs (in cluster1). The mAP adopts a group contention algorithm in which subsets of dAPs opportunistically contend for netMIMO transmission. When dAP1 and dAP2 sense an idle channel, the mAP may choose to synchronize them into one group and then run a probabilistic contention algorithm to acquire the channel and avoid collision with external dAPs (those in different clusters). The mAP's choice between spatial reuse (schedule subset of dAPs for contention) and multiplexing gain (wait for all dAPs to become idle) is driven by a local optimization algorithm, which realizes proportionally fair channel access among all dAPs.

After a group of dAPs win channel contention, the mAP needs to match them with a set of clients. In particular, it needs to modulate their data and budget the transmit power of each dAP, so as to cancel cross-talk interference and maximize multiplexing gain. When the number of selected clients is less than the number of dAPs, the dAPs can beamform their power to the clients to improve their SINR, *i.e.*, harvesting the diversity gain. These operations are managed by the mAP's PHY module, which optimizes long-term throughput subject to proportional fairness among clients.

# 4. EFFICIENT NEMOx CHANNEL ACCESS

## 4.1 Creating virtual APs

While allowing for opportunistic grouping of dAPs strikes a balance between their multiplexing gain and spatial reuse, there still remains an important question: *how is a dAP group formed?* Recall that this problem involves a conflict between asynchronous contention and synchronized cooperation between dAPs. NEMOx resolves this tension by separating contention from dAP grouping. An mAP first prescribes all potential groups of dAPs in its cluster; each group is called a *virtual AP* (vAP). It then runs a contention algorithm that allows the vAPs to contend with each other and also with external vAPs.

A vAP decides on an idle channel only if all its dAPs sense idle channels, and switches to busy if any of its dAPs becomes busy. Upon sensing an idle channel, the vAP maintains a single backoff timer whose expiration triggers the netMIMO transmission of all dAPs contained in the vAP. As these dAPs are virtually synchronized within a vAP, but vAPs contend with each other asynchronously, we call this *semi-synchronized contention*. NEMOx's unique architecture makes it possible to realize such a contention

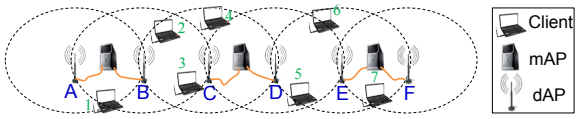


**Figure 5: An example topology in NEMOx with 3 clusters each containing 2 dAPs. The dotted circle represents a dAP's carrier sensing range.**

mechanism, because *the MAC/PHY operations of all vAPs and dAPs inside each cluster are centrally executed by the mAP.*

**Dominance relation.** In its simplest form, the vAPs in a cluster are equivalent to all of its subsets of dAPs. But direct enumeration of all subsets results in an exponential number of vAPs. Since not all cooperating sets contribute equally to the cluster's capacity, NEMOx discriminates them by defining a *dominance* relation.

A vAP $S_i$ is said to *dominate* another vAP $S_j$ within the same cluster if $S_j$'s dAPs is a subset of $S_i$'s, and $S_i$'s interfering dAPs in neighboring clusters are a subset of $S_j$'s dAPs. In this case, $S_i$ has similar or fewer number of contenders than $S_j$, but it has a higher multiplexing gain. The dominated vAPs in $S_j$ are pruned to achieve higher computation and contention efficiency. For example, for the leftmost cluster in Figure 5, the vAP $\{A, B\}$ dominates $\{B\}$ because they are interfered by the same set of dAPs (*i.e.*, dAP $C$) in other clusters, and $\{B\}$ is only a subset of $\{A, B\}$.

To establish a dominance relation, vAPs simply sample the channel status, instead of decoding the identities of their interfering dAPs—which is not always feasible. If vAP $S_j$ is always busy when $S_i$ is busy, but not vice versa, then $S_i$'s interfering dAPs are a subset of $S_j$. The mAP performs such sampling and refreshes the dominance relation periodically (akin to an 802.11 beacon period), which can be done easily as all vAPs' operations are running centrally at the mAP.

## 4.2 Semi-synchronized CSMA

By constructing vAPs, an mAP translates the medium access problem of an entire cluster into channel contention between vAPs. Below we describe the channel-access protocol from the perspectives of an mAP and its vAPs.

**Client association.** If a client associated with the cluster has downlink packets to receive, then all dAPs which can deliver packets to the client will be marked as potential transmitters. Further, all vAPs that contain any potential transmitters, will trigger channel contention. Depending on the outcome of contention, a client may be served by a different vAP for each transmission attempt. Such a *soft association* mechanism is a unique feature of the NEMOx architecture, as dAPs or vAPs in a cluster are centrally controlled by the mAP, and can flexibly adjust their association with clients on a per-packet basis. This flexibility makes it possible to vary the throughput provisioning for a client by matching it with vAPs of different sizes, which will be further optimized in NEMOx's PHY component (see Sec. 5).

**State transition.** NEMOx runs a probabilistic channel-access algorithm, where each vAP adjusts its aggressiveness in contention following the state-transition diagram in Figure 6. A vAP updates its state on a per-time-slot basis. The slot duration is the same as that in 802.11 (*e.g.*, 9$\mu s$ in 802.11g). Time slots are synchronized only for vAPs within the same cluster. A vAP, $V_j$, starts with the *busy state* by default. It enters the *idle state* if all of its dAPs sense an idle channel for a fixed number of slots (the DIFS period defined in 802.11 [9]). Then, it transits to *contention state* with a *contention probability* $q_j$. With probability $(1-q_j)$, it remains in the idle state, and advances its contention probability as:

$$q_j \leftarrow q_j + \alpha \quad (1)$$

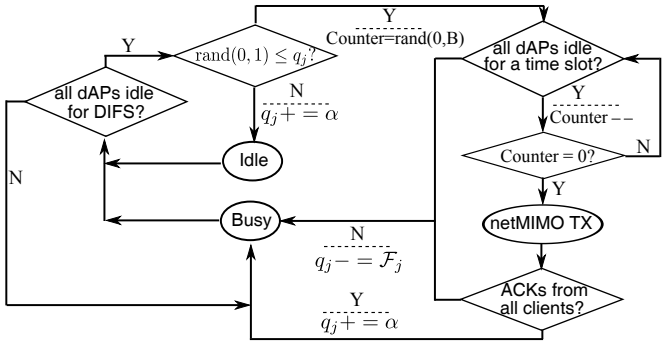


**Figure 6: State-transition diagram for a vAP.**

where $\alpha \in (0,1)$ is a fixed parameter that determines how aggressively $V_j$ should advance its contention probability.

After entering the contention state, it starts the backoff procedure by selecting a random number between 0 and the backoff window $B$, and counting the corresponding number of idle slots. If the backoff counter expires on time, $V_j$ calls its PHY module to start netMIMO transmission. If $V_j$ senses a busy channel during backoff due to another (either internal or external) vAP acquiring the channel beforehand, then it declares a *contention failure*, switches back to busy state, and reduces its contention probability as:

$$q_j \leftarrow q_j - \mathcal{F}_j \tag{2}$$

where $\mathcal{F}_j$ is a parameter to be optimized (see Sec. 4.3.3).

Upon completing transmission and receiving ACKs from all intended clients, $V_j$ returns to its initial busy state and advances its contention probability according to Eq. (1). Otherwise, a *transmission failure* occurs and $V_j$ reduces its contention probability according to Eq. (2).

In NEMOx, collision can only occur between vAPs of different clusters. When vAPs inside the same cluster fire backoff counters simultaneously, the mAP randomly enables one of them. By migrating the medium access from node (dAP) level to group (vAP) level, NEMOx substantially reduces the number of contention domains compared to traditional 802.11 WLANs, thus lowering the collision overhead.

It is now evident that the key to NEMOx's MAC efficiency and fairness lies in how a vAP advances or regresses its contention probability $q_j$ as in Eqs. (1) and (2). In what follows, we forward-engineer these operations to achieve optimal and fair channel access among all dAPs.

## 4.3 Optimizing medium access

As clients' population is dynamic and their traffic demands are unpredictable, NEMOx opts to provision *optimal and fair channel access* for the netMIMO infrastructure, *i.e.*, the dAPs instead. Responsibility for fairness among clients is delegated to the PHY module (Sec. 5) which selectively serves clients, given the dAPs available for netMIMO transmission.

### *4.3.1 Formulation of the optimization problem*

The overarching goal of the MAC module is thus to optimize the semi-synchronized contention algorithm, run by vAPs, to ensure optimal and fair channel access for all dAPs—the actual transmitters. Suppose $r_i$ is the probability of successful transmission for dAP $i$, and $\mathcal{C}$ the set of clusters. Then, the optimization problem can be cast as:

$$\max \quad \sum_{K\in\mathcal{C}} \sum_{i\in K} \alpha_K U(r_i). \tag{3}$$

Each dAP $i$ is associated with a utility function $U(r_i)$, which reflects the fairness objective. When $U(r_i) = \log(r_i)$, maximizing the sum utility is proven to achieve global optimality *w.r.t.* $r_i$ and proportional fairness [12]. $\alpha_K$ is the weight that can be used for prioritizing cluster $K$.

The optimization objective must be pursued subject to network resource constraints. The nature of CSMA contention allows at most one vAP to transmit in any contention domain, which can be characterized by a maximal clique—a complete graph in which there exists an edge between two vAPs if they can sense each other. In other words, the sum contention probability of all vAPs in any maximal clique should not exceed 1. Let $\mathcal{M}$ denote the set of maximal cliques in the network, then

$$\sum_{j\in m} q_j \le 1, \quad \forall m \in \mathcal{M}. \tag{4}$$

Furthermore, the effective transmission probability, $r_i$, of dAP $i$, equals the sum transmission probability of all vAPs containing it. Let $P_j$ be the probability that vAP $j$ experiences a transmission/contention failure, then its effective transmission probability is equal to $(1-P_j)q_j$. So, we have:

$$r_i = \sum_{i\in j} (1-P_j) q_j \tag{5}$$

Note that $r_i$ is the probability of successful transmission and $q_j$ is the overall *contention probability*. To derive $r_i$ from $q_j$, an attrition factor $(1-P_j)$ is multiplied, thus accounting for both transmission and contention failure.

Consequently, the semi-synchronized contention algorithm can achieve optimality and fairness if it solves the optimization problem (3) subject to constraints (4) and (5). With this formulation, the rate of each dAP would be a function of all of the vAPs it belongs to, as well as their collision probability. Moreover, with the utility defined *w.r.t.* each dAP, the above optimization implicitly favors vAPs with more dAPs and hence a higher netMIMO gain.

The solution of this optimization directly provides the channel contention probability $q_j$ for each vAP $j$. But solving the problem is non-trivial as it involves exchange of global information between all vAPs and dAPs in a network, which we strive to avoid. In what follows, we derive a decentralized solution that can be mapped to each vAP's local state transitions (Figure 6) but can still achieve the optimization objective.

### *4.3.2 Decentralized solution for each vAP*

We first obtain a decentralized solution to the optimization problem (3) by decomposing its variables into subsets, each associated with one NEMOx cluster. Observing that (3) is a concave optimization problem, we can obtain its Lagrangian as:

$$L = \sum_{K\in\mathcal{C}} \sum_{i\in K} \alpha_K U(r_i) - \sum_{m\in\mathcal{M}} \sum_{j\in m} \beta_m q_j \tag{6}$$

where $\beta_m$ is the Lagrange multiplier for maximal clique $m$. We can write $\beta_m$ as $\beta_m = \beta P_m$, where $P_m$ is the collision probability in maximal clique $m$. Since a vAP $j$ may belong to multiple maximal cliques, its collision probability is $P_j = 1 - \Pi_{m:j\in m}(1-P_m)$. If $P_m$ is kept small, then we have:

$$P_j \approx \sum_{m:j\in m} P_m \tag{7}$$

The assumption of a small $P_m$ requires that the collision probability of any contention domain within the network be kept low. This can be satisfied by keeping a reasonably small value for $\alpha$ and a large value for $B$ (Figure 6), so that no vAP becomes overly aggressive during contention. A similar assumption has been used in traditional utility-based wireless MAC design [10].

Following the approximation in Eq. (7), the parameter $P_j$ can be approached by allowing vAP $j$ to locally keep track of its con-

tention/transmission failure probability, without knowledge of its maximal cliques or loss probabilities in each of them. Now, substituting for $\beta_m$ and making the approximation, we have $L = \sum_{K \in \mathcal{C}} L_K$, where

$$L_K = \sum_{i \in K} \alpha_K U(r_i) - \beta \sum_{i \in K, j: i \in j} P_j q_j. \quad (8)$$

The analysis in [10, 11] has shown that the aggregate utility for a system of equations of the above form is maximized when the individual components maximize their own utilities. While the component corresponds to a single link in traditional WLANs, this corresponds to a cluster $K$ in netMIMO, thereby requiring joint adaptation of vAPs within each cluster for optimality. Applying the KKT conditions for optimality with respect to each vAP $j$, we obtain

$$\frac{\mathrm{d}L_K}{\mathrm{d}q_j} = \sum_{i \in j} \alpha_K U'(r_i)(1 - P_j) - \beta P_j = 0, \ \forall j \in m. \quad (9)$$

Since $U(r_i) = \log(r_i)$, at the optimum, for every vAP $j$ in cluster $K$, we have:

$$\frac{\mathrm{d}L_K}{\mathrm{d}q_j} = \sum_{i \in j} \alpha_K (1 - P_j) r_i^{-1} - \beta P_j = 0. \quad (10)$$

$$\text{Or equivalently, } \alpha_K (1 - P_j) - \beta P_j \cdot \Big( \sum_{i \in j} (1/r_i) \Big)^{-1} = 0.$$

While $\alpha_K$ can be used to prioritize different clusters, we set $\alpha_K = \alpha'$ for an equal bias. Given that $P_j$ can be locally inferred by each vAP $j$, the above optimality condition can be approached by adapting each vAP $j$'s contention probability in a completely decentralized manner as:

$$q_j \leftarrow q_j + \alpha' - P_j \Big( \beta (\sum_{i \in j} (1/r_i))^{-1} + \alpha' \Big). \quad (11)$$

The adaptation mechanism for each vAP follows a gradient approach based on the KKT conditions. Further, since the utility function $U(r_i) = \log(r_i)$ and the resulting Lagrangian are concave with respect to each $q_j$, there exists a unique maximum, to which the individual adaptations converge. We omit the detailed proof of convergence as it bears a spirit similar to the one in [10].

### 4.3.3 Mapping optimization solution to NEMOx

We can now map the optimization solution (11) to NEMOx's channel-access algorithm shown in Figure 6. The adaptation can be done in discrete time frames, each spanning one of three states: idle, transmission success, and transmission/contention failure. At the end of each frame, a vAP $j$ advances its contention probability by $\alpha'$, so $\alpha$ in NEMOx (Eq. (1)) should be set to $\alpha'$. A large value of $\alpha'$ leads to fast adaptation, but cause oscillation of $p_j$ around its optimum. By default, we set $\alpha' = 0.05$. By using (11), a vAP should deduct its $q_j$ by $(\beta(\sum_{i \in j}(1/r_i))^{-1} + \alpha)$ whenever a transmission/contention failure occurs (with probability $P_j$). Since failure is conditioned upon a contention attempt with probability $q_j$, the actual deduction should be scaled down by $q_j$. The resulting deduction is exactly the parameter $\mathcal{F}_j$ in Figure 6:

$$\mathcal{F}_j = q_j^{-1} \Big( \beta (\sum_{i \in j} (1/r_i))^{-1} + \alpha \Big). \quad (12)$$

When following the adaptation step of (11), each vAP $j$ should truncate $q_j$ to make it fall in the range $[0, 1]$.

Note that the parameter $\beta$ represents a scalar component for the Lagrange multiplier component $\beta_m$ in Eq. (6). A larger $\beta$ incurs a larger penalty to $p_j$ when collision occurs, but larger oscillation around the optimum. By default, we use an empirical value of $\beta = 0.25$ which keeps a relatively low collision probability.

When mutually interfering vAPs advance their contention probabilities to high values, severe collision may occur which will waste channel time for corrupted packets. NEMOx reduces the risk of collision with an additional backoff, which buffers and randomizes vAP's transmissions. The backoff window size $B$ need not be adapted over time as a vAP can react to and reduce collisions by adapting its contention probability. Here $B$ is defaulted to 32.

# 5. NEMOx: NETMIMO TRANSMISSION

NEMOx's PHY module resolves a second challenge unique to netMIMO, *i.e.*, optimizing power budgeting for dAPs and ensuring fairness among clients. This section describes the PHY module from the perspective of a vAP (and its dAPs) that has won channel contention. We first introduce how the vAP budgets power for a given set of clients, and then discuss how it selects clients for fairness.

## 5.1 Optimal power budgeting for dAPs

As we observed in Sec. 3, a direct application of ZFBF precoding to netMIMO may severely degrade its performance. The fundamental reason for this is that ZFBF forces dAPs to explore multiplexing gain and implicitly allocates the same power for all clients [7]. NEMOx overcomes this limitation with optimal power budgeting. It trades multiplexing gain for diversity gain in unbalanced topologies, by beamforming multiple dAPs' power to one client, while also opportunistically scheduling clients to reduce topology imbalance. The tradeoff between diversity and multiplexing gains is made automatically via the following optimization framework that performs joint precoding and power (JPP) allocation:

$$\text{JPP:} \quad \max \quad \sum_{i=1}^{|D|} w_i \log(1 + \frac{P_i}{N_0}) \quad (13)$$

$$\text{s.t.} \quad P_i = |\sum_{k=1}^{|S|} h_{ik} v_{ki}|^2, \forall i \in D \quad (14)$$

$$\sum_{i=1}^{|D|} |v_{ki}|^2 \leq P_{\max}, \forall k \in S \quad (15)$$

$$\sum_{k=1}^{|S|} h_{jk} v_{ki} = 0, i \in D, j \neq i, \quad (16)$$

where $D$ and $S$ are the set of clients and dAPs, respectively. Eq. (14) represents client $i$'s received power under the joint effects of precoding $v_{ki}$ and channel distortion $h_{ik} (k \in S, i \in D)$ *w.r.t.* dAP $k$. Eq. (15) is the per-dAP power budget constraint. Eq. (16) represents the precoding constraint, *i.e.*, precoded data symbols intended for client $i$ should cancel each other when arriving at client $j$ ($i \neq j$) after experiencing channel distortion. The weight $w_i$ associated with client $i$ determines throughput fairness and will be discussed in Sec. 5.2.

The objective of this JPP framework is to maximize the weighted sum throughput of all clients, *by optimizing elements of the precoding matrix $v_{ki} (k \in S, i \in D)$, which in turn determines dAP $k$'s power budget for client $i$.* When the topology is unbalanced, this objective implicitly favors a few clients with high capacity, instead of fairly serving all clients with extremely low capacity. To the best of our knowledge, such a tradeoff between diversity and multiplexing gain has not been addressed in existing netMIMO system designs [5, 6]. The significance of both the problem and our approach will be highlighted in Sec. 7.1 through our testbed experiments.

The JPP formulation can be easily proven to be non-convex [13] *w.r.t.* the real and imaginary components of $v_{ki}$, due to the norm operator in Eq. (14). Fortunately, by phase-shifting the vector $v_{ik}, \forall k \in S$ appropriately, we can restrict $Im(\sum_{k=1}^{|S|} h_{ik} v_{ik}) = 0$, while both constraints (15) and (16) are insensitive to the phase-shift. The resulting problem then becomes convex, and can be easily solved using standard convex optimization techniques.

**Algorithm 1** Opportunistic client scheduling.

1. **Input:** $S$: set of dAPs in a vAP who won contention; $D$: set of clients that can be served by the dAPs in $S$.
   $\mathcal{D}$: set of clients in this cluster.
2. **Initialize:** $D' \leftarrow$NULL, $O_{\max} \leftarrow 0$
3. **while** $|D'| < |S|$
4. **foreach** $i \in D \backslash D'$
5. Solve JPP, with the set of dAPs $S$ and set of clients $i \cup D'$. Obtain optimal sum capacity $O_i$.
6. **if** $O_i > O_{\max}$ **then** $O_{\max} \leftarrow O_i$; bestClient$\leftarrow i$;
7. **endif**
8. **endfor**
9. $D' \leftarrow$ bestClient $\cup D'$
10. **end while**
11. Run netMIMO transmission from $S$ to $D'$.
12. Client $i$ throughput is $R_i$ in this transmission, $i \in \mathcal{D}$
13. $\overline{R}_i \leftarrow \gamma R_i + (1-\gamma)\overline{R}_i$
14. $w_i \leftarrow 1/\overline{R}_i$

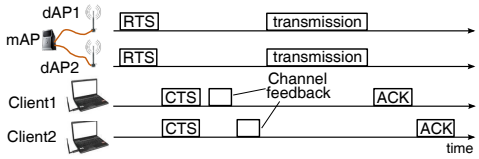


**Figure 7: Flow of operations for a netMIMO transmission in NEMOx.**

## 5.2 Opportunistic client scheduling

NEMOx adopts an opportunistic client scheduling algorithm (Algorithm 1) to restrict the set of clients to be served in each transmission attempt, and integrates client selection with the JPP precoding matrix design to achieve its fairness objective.

Client selection affects the topology pattern (balanced or unbalanced), which in turn affects the netMIMO capacity. This observation leads to an iterative client-selection algorithm for NEMOx. In each iteration, the algorithm searches for a *bestClient* that maximizes the weighted sum rate (according to JPP), when served together with those clients already selected. The selection aborts if the number of selected clients exceeds the number of dAPs, or the bestClient yields a lower sum rate than that in the previous iteration.

Then, netMIMO transmission is executed and client $i$ achieves a throughput of $R_i$. It then updates its time-averaged throughput $\overline{R}_i$ using a moving average with a smoothing factor $\gamma$ (set to 0.1 by default). Further, its throughput weight $w_i$ is adjusted to achieve a certain long-term fairness objective. For *proportional fairness*, we can configure $w_i$ to be the inverse of $\overline{R}_i$ [14].

## 5.3 Channel estimation, reservation and ACK

Figure 7 illustrates a typical flow of operations in a netMIMO transmission, which involves a set of dAPs selected by NEMOx's MAC module, and clients selected by the above opportunistic client scheduling algorithm.

Before starting the data transmission, the dAPs synchronously broadcast the same RTS packet that indicates the transmission duration and clients' addresses. Then, all selected clients return the same CTS packet. The CTS packet contains the duration of this transmission attempt, and is sent by all clients synchronously to reserve a channel from transmitters in neighboring clusters. The dAPs' RTS packets serve as a reference broadcast for all clients, so that they can synchronize their CTS responses. Since the CTS packets contain the same information, they do not collide with each other when synchronized. Such synchronized responses are shown

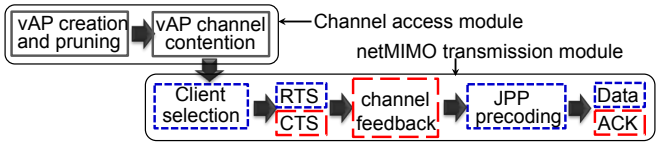


**Figure 8: Summary of NEMOx operations.**

to be effective even when COTS WiFi and ZigBee hardware are used [15, 16].

After the RTS/CTS handshake, the clients sequentially feed back the channel state information (CSI), *i.e.*, the channel vector from the dAPs to itself. Such feedback is necessary for the mAP to compute the precoding matrix following JPP. The netMIMO transmission is aborted if feedback from some clients is missing, perhaps due to a CTS channel reservation failure. In such a case, other vAPs can reuse the transmission opportunity, thanks to NEMOx's contention-based channel access protocol. If the netMIMO transmission succeeds, an ACK is also sent sequentially by each client in a similar manner to the CSI feedback.

NEMOx's CSI and ACK feedback operations are similar to those in 802.11ac, but its channel reservation mechanism is unique. In 802.11ac's MU-MIMO mode [9], an AP needs to broadcast a *TXOP* packet, which signals neighboring transmitters and reserves the channel. TXOP itself does not resolve the hidden terminal problem; NEMOx addresses it using the synchronous CTS from clients.

NEMOx's client-selection algorithm needs a channel matrix from the dAPs to all potential clients (line 5 in Algorithm 1). Estimating such a channel matrix incurs huge overhead. Observe, however, that dAPs and clients tend to spread over a large area, and small-scale fading (which causes small magnitude and phase variation) has relatively small impact on topology imbalance, compared to large-scale pathloss due mainly to transmitter/receiver distance. Therefore, without losing much accuracy, NEMOx runs client selection using the latest channel gain statistics in previous transmissions, so as to reduce the channel-estimation overhead.

# 6. IMPLEMENTATION

**Flow of Operations in NEMOx.** Figure 8 summarizes NEMOx's flow of operations throughout one channel contention and transmission attempt. An mAP periodically runs the vAP prunning algorithm (Sec. 4.1). It continuously contends for channel access on behalf of all remaining vAPs (Sec. 4.2). The vAP that wins contention in each cluster will start its netMIMO transmission attempt immediately. It first uses the client selection algorithm (Sec. 5.2) to determine the set of clients to serve, and initiates the RTS/CTS exchange with the selected clients and obtains channel matrix (Sec. 5.3). Then, it computes the precoding matrix $\boldsymbol{v}$ following the JPP framework (Sec. 5.1), precodes the data vector $\boldsymbol{x}$ into a new vector $\boldsymbol{vx}$, and sends each element of the vector through the corresponding dAP. Based on ACK feedback the clients, the vAP infers transmission success/failure, updates contention parameters and starts a new round of contention (Sec. 4.2).

**Implementation.** We have implemented a prototype of NEMOx on the WARP [8] software radio platform. We use the original WARP board as an mAP, and deploy the dAPs by extending the WARP antennas (with a radio board) by up to 30 ft using LMR-400 50Ω coaxial cables and SMA male-to-female connectors (Figure 9). Based on the WARPlab driver, we have implemented a full-fledged multi-user MIMO-OFDM modulation/demodulation library to support NEMOx. Figure 10 shows the basic components and interfaces in our implementation.

*Transmit path.* In the transmit path, we implement the NEMOx client-selection algorithm to determine the set of dAPs/clients to

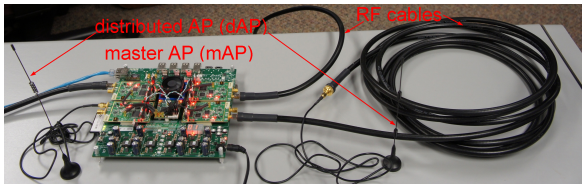


**Figure 9: Building a NEMOx cluster using WARP.**

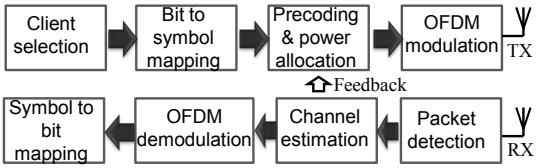


**Figure 10: The MIMO-OFDM PHY layer implementation of NEMOx on WARP.**

be used for netMIMO transmissions. Then, the digital bits of each client are mapped to symbols via BPSK. NEMOx's joint precoding and power-allocation algorithm is then performed in the frequency domain (on the symbols carried by each OFDM subcarrier). The precoded symbols for each dAP are modulated using OFDM and sent over the air.

*Receive path.* Each client runs the receiver path that detects the packets and then estimates the channel from each dAP. All packets start with a short-preamble that can be detected using a self-correlation based algorithm [17]. Due to the long interface latency between the WARP radio and its PC host, the synchronous CTS feedback from clients cannot be directly implemented. Thus, we simulate the RTS/CTS-based channel reservation with a C++ based module running on the PC host, as will be detailed below. Our implementation of channel estimation follows a similar procedure as in 802.11ac. Specifically, the dAPs first send the same short-preamble synchronously, and then sequentially send a long-preamble, which is used for each client to correct frequency offset and estimate the CSI. The CSI is estimated for each OFDM subcarrier and contains the channel phase/amplitude distortion between all dAPs and the receiver itself. CSI elements are quantized to 8-bit following 802.11ac [9] and fed back to the mAP for precoding. In our implementation, all dAPs' and clients' signal processing operations run on a PC host, so the feedback is realized directly using function calls. This alleviates the problem of CSI feedback delay, although the signal processing delay is still on the order of tens of milliseconds, much longer than an actual hardware implementation.

Following the long-preambles is an additional preamble sent by all dAPs concurrently and used by each receiver to estimate the composite channel created by channel distortion and precoding. Based on the estimation estimation results, the receiver can demodulate the OFDM symbols and decode the digital bits therein. Within each OFDM symbol, 4 pilot subcarriers (with known bits) are sent by the transmitter and used for correcting residual errors in the frequency offset estimation using long-preambles.

*Implementation of NEMOx MAC.* We implement NEMOx's MAC protocol as a C++ back-end module running on the WARP controller. Time-critical functionalities, such as backoff countdown and inter-frame timing, are realized using a virtual timer instead of wallclock time. Carrier sensing relation between dAPs are measured offline for each experimental topology and fed into the MAC module. If contention succeeds, the MAC of the winning mAP calls its PHY transmit path to realize the packet transmission. This approach circumvents the long latency of the WARP-PC interface, but preserves the contention behaviors of different mAPs, as well as the channel pathloss/fading effects.

# 7. EVALUATION

In this section, we first evaluate NEMOx's PHY components. Then, we run it in a multi-cluster testbed to evaluate its *network-level* throughput, fairness and scalability.

## 7.1 Micro-benchmark evaluation

We deployed a single cluster in an office environment (its floor map shown in Figure 11). To isolate ambient interference, all experiments were conducted at night, and over a 2.4 GHz ISM band (channel 14) unused by any other nearby devices. As a WARP board can only support up to 4 dAPs, we use trace-driven emulation for more than 4 dAPs. For emulation, each dAP broadcasts short probing packets containing only the synchronization and channel estimation preambles. Channel matrices are recorded by clients for 1 minute and then replayed in our PHY module implementation. Sampling one channel matrix takes only around 8.7 ms with the latest WARPLab driver, well below the typical coherence time for indoor radio environments (800 ms according to [18]) and ensures that the channel matrices represent a snapshot of the real channel effects. The accuracy of this emulation will be validated by comparing it with real-time experiments.

**Multiplexing gain.** We compare NEMOx with a baseline scheme in which dAPs are non-cooperative, referred to as *NonCoop*, essentially the default protocol in 802.11 WLANs. To isolate the MAC-layer effects, we first assume NonCoop can centrally schedule dAPs alternately (*w/o MAC*). By default, we run UDP transmissions with packet size 1.5KB and BPSK modulation. The SINR of *decoded* BPSK symbols is then mapped to the Shannon rate. Figure 12(a) plots NEMOx's multiplexing gain (reflected by per-client throughput gain) over NonCoop, where the number of clients increases along with that of dAPs. As dAP density rises, the number of concurrent transmissions (and hence multiplexing gain) grows almost linearly. Ideally, the gain should increase in the same order as dAP density, but the experimental results are shown to be less than ideal, *e.g.*, the average capacity gain is 6.45 with dAP density equal to 8. This is because the dAP-to-client channels are not perfectly orthogonal. Each dAP needs to reduce power to combat the channel correlation and cancel cross-talk interference. Similar observations have been made in an existing theoretical analysis [14].

Figure 12(a) also compares the trace-driven emulation (for dAP density 2—4) with real-time experiments (represented with solid symbols), which match each other very well. Our analysis of the traces show that the received signal power variation is negligible over tens of seconds to several minutes, as our experimental setting avoids ambient mobility and interference. Thus, collecting and replaying channel traces produces consistent experimental results with real-time runs.

We further enable NEMOx's MAC implementation and compare it with our implementation of 802.11 CSMA for NonCoop (*w/ MAC*). Figure 12(a) shows that NEMOx's gain grows to 3.92 and 7.91, for dAP density 4 and 8, respectively, even higher than w/o MAC. Under a realistic CSMA-based MAC, the NonCoop APs need to spend a substantial amount of time for channel contention and collision resolution. By contrast, NEMOx eliminates such overhead, and therefore, *NEMOx's multiplexing gain comes from both netMIMO cooperation at PHY layer and reduced contention overhead at MAC layer*. The combined capacity gain can be even larger than the dAP density in one cluster.

**Diversity gain.** When the client population is small, multiple dAPs can harvest diversity gain by beamforming the same data towards one client. Figure 12(b) evaluates such diversity gain for two clients: client A, located at the network edge, and client B, located near the network center (with similar distance to all dAPs). The for-

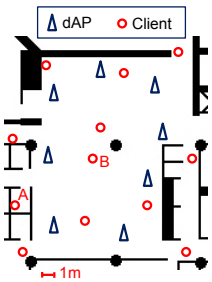


**Figure 11: Single-cluster testbed. Black dots are pillars in our building.**

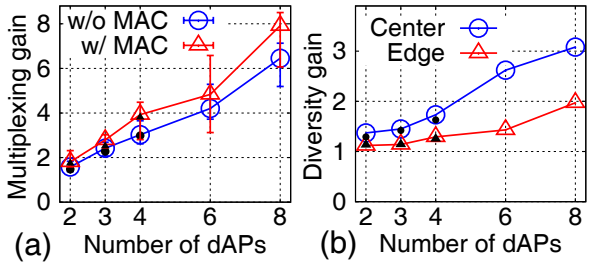


**Figure 12: NEMOx's multiplexing and diversity gains over non-cooperative multi-AP networks. Error bars represent maximum and minimum among clients.**

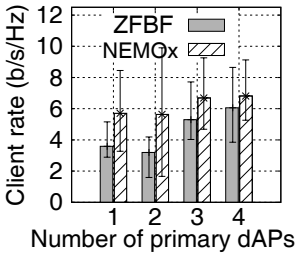


**Figure 13: Impact of power allocation: error bars show max and min of 5 topologies.**

mer benefits little from diversity, as its signal power mainly comes from the few dAPs nearby. The latter's capacity scales with the number of beamforming dAPs. With 8 dAPs, the average diversity gain is 3.08. This implies that the diversity gain scales sub-linearly with $M$ (the number of dAPs), consistent with the theoretical rule of $\log(M^2)$ scalability [14].

**Impact of power allocation.** NEMOx's JPP module is critical when the cluster's topology is unbalanced. In practice, such topologies are formed when clients are concentrated around a few dAPs, which we call *primary dAPs*. In Figure 13, we place 4 dAPs in a cluster, and vary the clients' concentrated locations to create different numbers of primary dAPs. The results show that, compared to ZFBF, NEMOx improves the mean link capacity by around 59% when the clients are concentrated near 1 or 2 dAPs. Despite its remarkable performance for co-located antenna systems [7, 19], the equal power allocation strategy in ZFBF may severely underutilize netMIMO's capacity in such topologies.

**Impact of opportunistic client selection.** An additional dimension of optimization in NEMOx is client selection. Figure 14 shows the experimental results when the number of dAPs is fixed at 4 while varying that of clients. Compared to a naive scheme that randomly selects 4 clients, NEMOx achieves a 58.9–75.6% higher average rate for clients. Even when the number of dAPs equals that of clients, the random scheme always schedules all clients. By contrast, NEMOx may partition the clients into multiple (overlapping) groups, each having a high sum-rate and served over different transmission attempts to achieve higher network capacity.

Figure 14 also plots the results from NEMOxZ, an alternative scheme that runs NEMOx's iterative client selection (Algorithm 1), but replaces the JPP component with ZFBF. The results show that NEMOxZ achieves performance comparable to NEMOx when the number of clients is much larger (*e.g.*, twice) than that of dAPs. This is because with more clients, NEMOxZ has a better chance to group clients that form a balanced topology, thus maintaining high capacity. Therefore, *when the number of clients is large, NEMOx can replace its JPP module with ZFBF to reduce its computational complexity.*

## 7.2 Scaling netMIMO with NEMOx

**Throughput optimality and fairness in benchmark topologies.** We first benchmark NEMOx's MAC-layer performance by deploying 3 clusters (top 3 ones in Figure 15), each containing 2 dAPs serving 2 clients. Every two adjacent dAPs can sense each other. The topology is essentially the same as Figure 5. It has a well-defined optimal and fair allocation of access rate for all dAPs, which can be obtained by solving the optimization problem (3). Our experiments focus on NEMOx's MAC-layer access opportunity and fairness, and isolate PHY-layer effects by forcing BPSK modulation (identical data rate) for served clients. We will later evaluate the joint MAC/PHY effects. We evaluate MAC throughput by sending saturated UDP traffic to each client, with maximum aggregated frame size of 4 KB, unless noted otherwise. We use 50ms as the CSI feedback period, assuming this is much shorter than the channel coherence time.

We compare NEMOx with 3 other schemes. The first is Oracle that directly assigns the optimal channel access probability $q_j$ to each vAP $j$. The second is FullCoop that groups all dAPs in each cluster into one vAP, *i.e.*, it greedily employs netMIMO cooperation gain without the semi-synchronized contention in NEMOx. FullCoop represents an extension of existing netMIMO proposals [5, 6] to multiple contention domains. The third is NonCoop representing the 802.11 MAC which runs CSMA among dAPs, *i.e.*, no netMIMO gain.

Figure 16 plots the throughput of each client. NEMOx achieves a similar level of total network throughput and per-client throughput compared to the Oracle MAC, implying that its distributed MAC approximates a centralized protocol well. FullCoop loses spatial reuse for dAPs 1 and 6 (which should have more transmission opportunities due to less contention), thus reducing the corresponding clients' throughput by more than 79%. Though NonCoop leverages spatial reuse, it sacrifices cooperation gain between dAPs, resulting in 32% lower network throughput than NEMOx in this benchmark topology. Note that for clients 2 to 5, FullCoop and NonCoop may achieve higher throughput than NEMOx. But this comes at the cost of proportional fairness for clients 1 and 6, which should have achieved higher throughput when served by dAPs 1 and 6.

We proceed to quantitatively verify NEMOx's fairness, *i.e.*, whether it delivers throughput gains to all clients by combining its MAC and PHY modules. As NEMOx targets proportional fairness, the fairness metric should be the log-utility function $\sum_{k=1}^{D} \log(H_k)$, where $H_k$ is the throughput of client $k$). We evaluate NEMOx with 2 clusters (involving dAPs 1–4), 4 clusters (dAPs 1–8), and 6 clusters (dAPs 1–12) in the testbed topology (Figure 15). By varying the number of clusters, we can test NEMOx under different patterns and levels of inter-cluster contention. Due to limited hardware, the 6-cluster case is evaluated using trace-driven emulation.

Figure 17(a) shows that NEMOx's network utility is substantially higher than NonCoop, *i.e.*, it delivers throughput gain without sacrificing fairness. Figure 17(b) plots the per-client throughput gain over NonCoop. The results show that NEMOx's throughput gain is strictly higher than 1 for all clients, which further verifies its MAC-level fairness. Consistent with the above benchmark evaluation, FullCoop sacrifices spatial reuse for certain clients, rendering their throughput gain below 1. Interestingly, NEMOx's median throughput increases slightly with the number of clusters, even under a fixed dAP density. A close examination reveals that in large topologies, NonCoop often starves links that fall in the middle of others (the "flow-in-the-middle" anomaly in 802.11 [20]). Through tight coordination of dAPs within each cluster, NEMOx alleviates

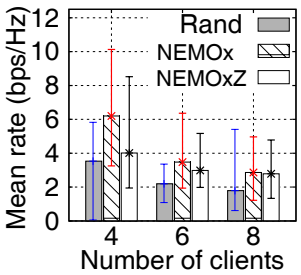


**Figure 14: Effectiveness of client selection.**

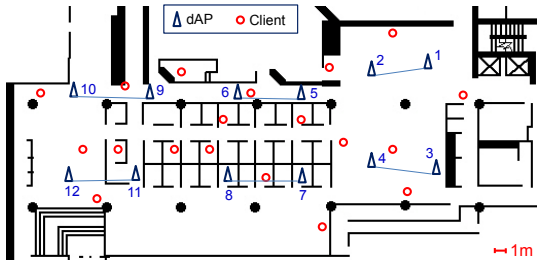


**Figure 15: Multi-cluster NEMOx testbed. Connected dAPs belong to the same cluster.**

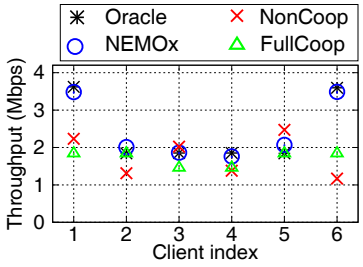


**Figure 16: Throughput optimality in a benchmark topology.**

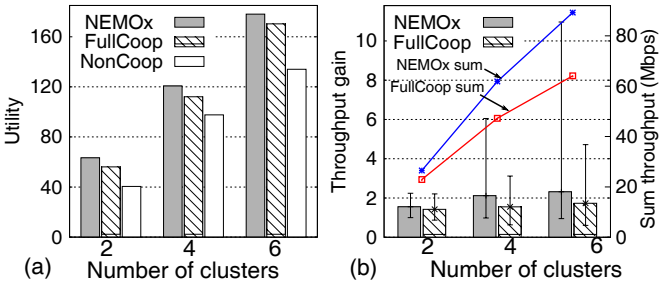


**Figure 17: Fairness evaluation. (a) Network utility; (b) Bars show the median of per-client throughput gain over non-cooperative multi-AP WLANs. Error bars show the min and max among all clients.**

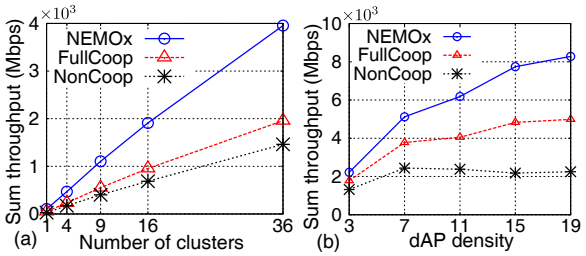


**Figure 18: Scalability with (a) network size (dAP density equals 3) and (b) dAP density (network size set to 4 clusters).**

such effects, thus substantially improving the throughput of starved clients. Figure 17(b) also implies that NEMOx's total network throughput scales as the number of clusters grows from 2 to 6.

**Scalability with network size.** We further verify the throughput scalability of NEMOx in large wireless networks with hundreds of nodes. Deploying such a network incurs formidable hardware cost. However, note that NEMOx's scalability mainly relies on its inter-cluster MAC, whose properties can be faithfully captured by the C++-based MAC implementation. We thus run the MAC together with empirical channel traces. We use a pair of WARP nodes to measure the channel magnitude and packet loss rate over 40 meters in our office building. The resulting empirical model is used to generate synthetic channel traces for given node locations, and fed into the WARPlab-based PHY and C++ based MAC modules to emulate NEMOx, FullCoop and NonCoop.

To evaluate NEMOx's scalability with network size, we increase the number of clusters from 4 to 36. Each cluster contains 3 dAPs uniformly placed around the cluster-center, and 6 clients randomly deployed within the coverage of dAPs. Each dAP uses a transmit power of 10 dBm. The dAPs' communication range *partially overlaps* with each other to ensure no dead spot, thus the network area expands with the number of clusters or dAPs. We define the communication range as the empirical distance within which a dAP can deliver at least 6 Mbps bit-rate with less than 10% loss rate. The dAPs' carrier-sensing threshold is set to -70 dBm, translating to around 1.2× the communication range under the empirical channel model. From the results in Figure 18(a), we find that all three schemes can scale their network capacity linearly with network size. However, NEMOx's scalability is higher by more than 2× (with a dAP density of only 3), as it opportunistically leverages spatial reuse and cooperation between dAPs, which are missing in FullCoop and NonCoop, respectively.

**Scalability with dAP density.** To validate NEMOx's scalability with dAPs' deployment density, we fix the number of clusters at 4, while increasing the number of dAPs per cluster from 3 to 19. Other experimental settings remain the same as above. The results in Figure 18(b) show that the capacity of both NEMOx and FullCoop increases with network density, due to more concurrency (multiplexing opportunities) brought by netMIMO. However, NEMOx's MAC can leverage the spatial reuse between dAPs, thus achieving a throughput gain of 1.33× FullCoop at a practical dAP density of 7, and 1.64× with density 19. It is clear that FullCoop, a straightforward extension of existing netMIMO schemes from single cluster (contention domain) to multiple clusters, cannot fully exploit the multiplexing gain of netMIMO. The random access MAC in NEMOx is critical to capacity scaling in this regard.

NonCoop's capacity increases initially due to higher spatial-reuse created by a denser topology. However, the gain soon plateaus when interference between links dominates. NEMOx achieves a throughput gain of 2.21× with dAP density 7, and the gain increases consistently with higher density. Thus, NEMOx makes the best balance between multiplexing within each cluster and reuse across clusters.

**Channel estimation overhead.** Like other MU-MIMO and netMIMO schemes, NEMOx's gain comes with MAC-layer overhead due to CSI feedback. In Figure 19, we evaluate the relative channel time cost of CSI feedback overhead in comparison with that of data transmissions. The results show that when the dAP density is high, more dAPs tend to be grouped for netMIMO transmission, thus raising the feedback overhead. However, even in the extreme case with an unusually high density of 19, the overhead is around 7%, negligible compared to NEMOx's throughput gain. Such overhead does not grow unbounded with dAP density, mainly because NEMOx's opportunistic MAC prevents the dAPs from greedily using large group sizes (Sec. 3.1.1). In addition, by increasing the packet size through frame aggregation, the overhead can be reduced to a minimum level (*e.g.*, below 2.2% for 4 KB packet size).

We remark that for effective precoding, the CSI feedback period must be shorter than the coherence time. Recent measurement studies of indoor radio environment [18, 21] found the channel coherence time to be on the order of several seconds for static nodes, and several hundred milliseconds for nodes moving at walking speed. These experiments were conducted under controlled settings, with

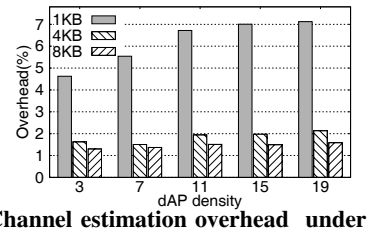


**Figure 19: Channel estimation overhead under varying dAP density and data packet size.**

short link distances and no moving objects around the links. Depending on the correlation threshold for defining coherence time, MIMO performance may be affected even when the coherence time is longer than the CSI feedback period [21]. In the above MAC-level experiments, we have used 50 ms as the CSI feedback period, which is conservative for static scenarios but may cause capacity loss for highly mobile scenarios. It has been observed in [18] that a feedback interval of 15 ms is sufficient for an indoor network environment with low-mobility and short-distance links. As we decrease the feedback interval from 50 ms to 15 ms, NEMOx's overhead would increase from around 7–23% (Figure 19). Its network throughput will be discounted by 16% accordingly, but still much higher than the above benchmark protocols.

# 8. DISCUSSIONS

**Compatibility with existing protocols.** NEMOx promotes cooperation between distributed APs. Hence, it is best applicable to managed network environments, such as conference venues, office buildings, and multi-media classrooms. Even after NEMOx is deployed in such an environment, legacy devices are unlikely to fade out anytime soon. Our main design choices for NEMOx were made with backward compatibility in mind. NEMOx' PHY layer is built on ZFBF, and its channel estimation and ACK mechanisms are consistent with the 802.11ac MU-MIMO standard. Its power allocation and client selection algorithms only customize the precoding matrix, and can be implemented in the driver. NEMOx's MAC uses a probabilistic algorithm and needs modifications to the 802.11ac AP. But it can be converted to a backoff-based algorithm by translating the contention probability into a backoff window size [11]. Moreover, NEMOx requires no modifications to clients. It can easily be degenerated to an 802.11-compatible network by regarding each dAP as the 802.11 AP and allowing the NEMOx mAP to run the 802.11 MAC for them. We plan to investigate the co-deployment of NEMOx and 802.11 networks in future.

**Uplink transmission.** In this paper, we focused on improving the downlink capacity of wireless networks using NEMOx, as downlink traffic is known to dominate wireless infrastructure networks. A simple way to accommodate uplink transmissions is to allow the clients to send RTS, and dAPs to defer the CTS, waiting for an opportunity when multiple uplink transmissions can be done simultaneously. Such uplink transmissions can exploit SIMO decoding algorithms [22] and will be utilized in our future work.

**Deployment issues.** We have built a netMIMO cluster by extending existing MIMO APs with commercial off-the-shelf RF cables (LMR-400). At 2.4 GHz, the cable causes an attenuation of 6.8 dB per 100 ft (sufficient to cover a typical indoor WLAN cell). With higher-quality cables, such as LMR-1700, the attenuation can be reduced to 1.7 dB, which is negligible and outweighed by NEMOx's cooperation gains.

**Multi-Antenna dAP and clients.** In NEMOx, each dAP or client has only one antenna. By allowing multiple antennas and integrating MU-MIMO communications algorithms, the per-client throughput can be boosted further. NEMOx's inter-cluster channel-access algorithm will still be valid in such a case, but the JPP frmework needs to be re-designed to exploit this capability.

**Dynamic environment.** Our experiments focused on static scenarios where channel state remains relatively stable, so as to prevent experimental artifacts caused by the latency of the WARPLab framework. In practice, network dynamics affect MIMO performance, and the effects depend on a variety of factors, such as node mobility, number of antennas, link distances, and the variation of LOS/NLOS. A comprehensive study of such dynamics and the impacts on NEMOx's performance would require a full-fledged real-time implementation of NEMOx and experimentation on a large-scale testbed with high dAP density. This is a matter of our future inquiry.

# 9. RELATED WORK

Network MIMO has recently attracted significant attention from the communications and information theory communities. Using simple channel models, existing work [23, 24] has proven that the capacity of a single-cluster netMIMO can scale with the number of distributed transmitters (or antennas). Analysis and simulation of multi-cluster netMIMO also verified its advantage in large cellular networks [25, 26]. Precoding and scheduling algorithms have been migrated from MU-MIMO to deal with problems in netMIMO [27], and have also been applied across netMIMO clusters [28, 29]. This line of research commonly assumes a cellular network model, where a netMIMO cluster is deployed in isolation (*e.g.*, leveraging dedicated spectrum) or neighboring clusters are synchronized and centrally managed through TDMA/FDMA. It is, however, difficult to apply such a model to the indoor multi-cell WLANs that are decentralized and self-organized via CSMA-based protocols.

Coordinated Multi-Point Transmission (CoMP), a concept built on the theoretical foundation of network MIMO, is expected to be realized in future LTE networks [2, 30]. CoMP allows netMIMO cooperation between base stations that form a cluster, so as to remove mutual interference. It is still in a conceptual development stage and there remain many practical challenges [31], *e.g.*, synchronization and scheduling of clusters. To date, only small-scale field tests [31, 32] have been done, while focusing on a single isolated cluster.

MU-MIMO and netMIMO communications for indoor WLANs have recently been realized in experimental settings. Both downlink [7] and uplink interference cancellation algorithms for MU-MIMO have been implemented [22, 33]. Shepard *et al.* [19] pushed the limit of MU-MIMO by building a *massive antenna system* with 64 transmit antennas that can serve tens of users simultaneously. It has also been proven [34] that netMIMO only needs an order-of-magnitude smaller number of antennas to achieve the same spectrum efficiency as MU-MIMO, primarily because of much shorter distance between transmit and receive antennas. Yet netMIMO incurs extra infrastructure cost in deploying the dAPs.

Practical netMIMO systems (*e.g.*, [5]) usually share the same communication algorithm (*e.g.*, ZFBF) with MU-MIMO, except that the antennas are from distributed transmitters. However, existing netMIMO systems [5, 6] assumed a single contention domain where every transmitter can hear and synchronize with others. A primary objective of NEMOx is to enable netMIMO in large-scale multi-cell WLANs spanning multiple contention domains.

NEMOx's PHY-layer JPP framework synthesizes linear precoding and power allocation, and is integrated with opportunistic user selection. These individual problems have been explored in existing simulation and analytical studies. For example, Yoo *et al.* [35] proved the asymptotic optimality of ZFBF precoding when combined with user selection and water-filling-based power allo-

cation, but under a sum-power constraint that is impractical for netMIMO [2]. Per-antenna power constraints were incorporated in [36], where precoding and power allocation are separated, and no user selection/scheduling is allowed. Alternative precoding schemes, *e.g.*, DPC and TH-precoding [14], can be used to replace the ZFBF in NEMOx. Evaluation of such schemes is left as our future work.

# 10. CONCLUSION

In this paper, we have proposed a novel system, called NEMOx, to exploit netMIMO gain for scalable performance in wireless networks. NEMOx organizes a network into multiple clusters, optimizes and performs netMIMO within each cluster containing distributed APs, and manages interference and reuse across clusters efficiently through a decentralized channel-access mechanism. Our prototype implementation and evaluation of NEMOx on networked WARP nodes have shown scalable netMIMO performance both within each cluster and across the network. These indicate NEMOx's potential for scaling the gains of netMIMO in wireless networks.

## Acknowledgement

We appreciate insightful comments and feedback from the shepherd, Lin Zhong, and the anonymous reviewers. The work reported in this paper was supported in part by the NSF under Grants CNS-1160775 and 1317411.

# 11. REFERENCES

[1] 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.

[2] 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. 9, 2010.

[3] S. Gollakota, S. D. Perli, and D. Katabi, "Interference Alignment and Cancellation," in *Proc. of ACM SIGCOMM*, 2009.

[4] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random Access Heterogeneous MIMO Networks," in *ACM SIGCOMM*, 2011.

[5] H. S. Rahul, S. Kumar, and D. Katabi, "JMB: Scaling Wireless Capacity With User Demands," in *Proc. of ACM SIGCOMM*, 2012.

[6] 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.

[7] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, "Design and Experimental Evaluation of Multi-user Beamforming in Wireless LANs," in *Proc. of ACM MobiCom*, 2010.

[8] A. Khattab, J. Camp, C. Hunter, P. Murphy, A. Sabharwal, and E. W. Knightly, "WARP: a Flexible Platform for Clean-Slate Wireless Medium Access Protocol Design," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, 2008.

[9] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std. 802.11ac Draft 3.0*, 2012.

[10] T. Nandagopal, T.E.Kim, X. Gao, and V. Bhargavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *ACM MobiCom*, 2000.

[11] J. W. Lee, M. Chiang, and R. A. Calderbank, "Optimal MAC design based on utility maximization: Reverse and forward engineering," in *IEEE INFOCOM*, 2006.

[12] 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. 237–252, 1998.

[13] A. Wiesel, Y. Eldar, and S. Shamai, "Zero-Forcing Precoding and Generalized Inverses," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, 2008.

[14] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[15] M. Kurth, A. Zubow, and J. P. Redlich, "Cooperative Opportunistic Routing Using Transmit Diversity in Wireless Mesh Networks," in *Proc. of IEEE INFOCOM*, 2008.

[16] P. Dutta, R. Musaloiu-E, I. Stoica, and A. Terzis, "Wireless ACK Collisions Not Considered Harmful," in *In HotNets-VII: The Seventh Workshop on Hot Topics in Networks*, 2008.

[17] X. Zhang and K. G. Shin, "Adaptive Subcarrier Nulling: Enabling Partial Spectrum Sharing in Wireless LANs," in *Proc. of IEEE ICNP*, 2011.

[18] R. Kudo, K. Ishihara, and Y. Takatori, "Measured Channel Variation and Coherence Time in NTT Lab," *IEEE 802.11-09/0161r1*, 2010.

[19] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: Practical Many-Antenna Base Stations," in *Proc. of ACM MobiCom*, 2012.

[20] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. Knightly, and M. Chiang, "Experimental Evaluation of Optimal CSMA," in *Proc. of IEEE INFOCOM*, 2011.

[21] X. Xie, X. Zhang, and K. Sundaresan, "Adaptive Feedback Compression for MIMO Networks," in *Proc. of ACM MobiCom*, 2013.

[22] 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*, 2009.

[23] R. Heath, T. Wu, Y. H. Kwon, and A. Soong, "Multiuser MIMO in Distributed Antenna Systems With Out-of-Cell Interference," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, 2011.

[24] J. Zhang and J. Andrews, "Distributed Antenna Systems with Randomness," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, 2008.

[25] S. A. Ramprashad, H. C. Papadopoulos, A. Benjebbour, Y. Kishiyama, N. Jindal, and G. Caire, "Cooperative Cellular Networks Using Multi-User MIMO: Trade-offs, Overheads, and Interference Control Across Architectures," *IEEE Communications Magazine*, vol. 49, no. 5, 2011.

[26] H. Huang, M. Trivellato, A. Hottinen, M. Shafi, P. Smith, and R. Valenzuela, "Increasing Downlink Cellular Throughput with Limited Network MIMO Coordination," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, 2009.

[27] H. Huh, A. M. Tulino, and G. Caire, "Network MIMO With Linear Zero-Forcing Beamforming: Large System Analysis, Impact of Channel Estimation, and Reduced-Complexity Scheduling," *IEEE Transactions on Information Theory*, vol. 58, no. 5, 2012.

[28] J. Zhang, R. Chen, J. Andrews, A. Ghosh, and R. Heath, "Networked MIMO with Clustered Linear Precoding," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, 2009.

[29] S. Kaviani and W. Krzymien, "Multicell Scheduling in Network MIMO," in *Proc. of IEEE GLOBECOM*, 2010.

[30] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno, "Coordinated Multipoint Transmission/Reception Techniques for LTE-Advanced," *IEEE Wireless Communications*, vol. 17, no. 3, 2010.

[31] Patrick Marsch (Editor) and Gerhard P. Fettweis (Editor), *Coordinated Multi-Point in Mobile Communications: From Theory to Practice*. Cambridge University Press, 2011.

[32] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H. P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated Multipoint: Concepts, Performance, and Field Trial Results," *IEEE Communications Magazine*, vol. 49, no. 2, 2011.

[33] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen, "Rate Adaptation for 802.11 Multiuser MIMO Networks," in *Proc. of ACM MobiCom*, 2012.

[34] H. Huh, G. Caire, H. Papadopoulos, and S. Ramprashad, "Achieving "Massive MIMO" Spectral Efficiency with a Not-So-Large Number of Antennas," *IEEE Transactions on Wireless Communications*, vol. 11, no. 9, 2012.

[35] T. Yoo and A. Goldsmith, "On the Optimality of Multiantenna Broadcast Scheduling Using Zero-Forcing Beamforming," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, 2006.

[36] K. Karakayali, R. Yates, G. Foschini, and R. Valenzuela, "Optimum Zero-forcing Beamforming with Per-antenna Power Constraints," in *IEEE International Symposium on Information Theory (ISIT)*, 2007.