Random Access Signaling for Network MIMO Uplink

Teng Wei and Xinyu Zhang
University of Wisconsin-Madison
twei7@wisc.edu and xyzhang@ece.wisc.edu

***Abstract*—Increasing popularity of mobile devices and upload-intensive applications is rapidly driving the uplink traffic demand in wireless LANs. Network MIMO (netMIMO) can potentially meet the demand by enabling concurrent uplink transmissions to an AP cluster (APC) comprised of multiple access points. NetMIMO's PHY-layer communication algorithms have been well explored, but the MAC-level signaling procedure remains an open issue: prior to uplink transmission, a group of clients must gain channel access, and ensure synchronization and channel orthogonality with each other. But such signaling is fundamentally challenging, because netMIMO clients tend to be widely distributed and may not even sense each other. In this paper, we introduce the first signaling protocol, called NURA, to meet the challenge. NURA clients employ a novel medium-access-signaling mechanism to realize group-based random access and synchronization, without disturbing ongoing uplink transmissions. The APC leverages a lightweight user-admission mechanism to group users with orthogonal channels (and hence high uplink capacity), without requiring costly channel-state feedback from all users. We have implemented NURA on a software-radio based netMIMO platform. Our experiments show that NURA is feasible, efficient, and can readily serve as the *a priori* signaling mechanism for distributed asynchronous netMIMO clients.**

I. Introduction

The population of mobile devices has been growing at a tremendous speed. The resulting surge of wireless traffic demand poses a looming challenge to today's network infrastructures. Though current wireless LAN (WLAN) deployment prioritizes downlink capacity, uplink traffic is envisioned to grow rapidly [1] due to the emerging upload-intensive applications, such as photo/video sharing, crowd-sensing and mobile cloud services. Deploying more access points (APs) in a region cannot satisfy the projected uplink traffic demand, since the inherent CSMA mechanism only allows a single transmission within one contention domain.

Uplink multiuser MIMO (MU-MIMO) overcomes the limitation, and enables concurrent transmissions from multiple clients to a multi-antenna AP [2]. In theory, the maximum number of concurrent clients can equal the number of antennas at the AP — a factor known as the network's *degrees of freedom* (DoF) [2]. Practical experiments have demonstrated the feasibility of uplink MU-MIMO [3]–[5]. Yet, existing studies have focused exclusively on an AP with co-located antennas, which still suffers from inter-cell contention and low spectrum efficiency in dense multi-cell networks.

Network MIMO (netMIMO), also called distributed MU-MIMO, represents a promising architecture to overcome the limitation [6]. It groups multiple distributed APs (*dAP*) in an AP cluster (*APC*), which acts as a giant MU-MIMO AP through tight synchronization and data sharing (Fig. 1). Existing netMIMO systems have addressed *downlink* transmissions [7]. Extending such solutions to uplink requires precise synchronization and centralized scheduling of clients, which is only suitable under static, managed settings.

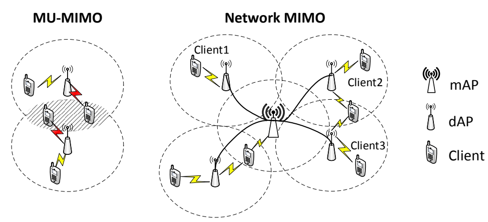

Fig. 1. Network MIMO (right) eliminates the inter-cell interference and improves coverage compared to traditional MU-MIMO (left).

We argue that a better way to bring netMIMO to WLANs' uplink is to enable random access for clients. Following a CSMA-like protocol, a client can contend for channel access at any time, and transmit a packet along with ongoing transmissions from other clients. Such a random-access protocol is commensurate with the distributed, asynchronous nature of legacy mobile devices with sporadic uplink traffic patterns. Concurrent uplink packets can be jointly decoded by the APC in a similar way as a conventional MU-MIMO AP decoder [3], [4], which eliminates interference between clients.

However, realizing such a random-access mechanism for uplink netMIMO entails fundamental *signaling* issues that do not exist in conventional random-access networks. *First*, before contending for channel access, a client needs to ensure the number of concurrent transmissions has not exhausted the APC's DoF. After contention, it also needs to know whether it wins. Such knowledge is readily available in conventional MU-MIMO networks [3], [5], [8] where clients can directly coordinate with each other. However, this no longer holds true for uplink netMIMO where the clients are widely distributed (spanning multiple conventional cells as shown in Fig. 1), and may not even hear each other. *Second*, for successful decoding, concurrent uplink transmissions must synchronize their symbol boundaries within a tolerable period called cyclic prefix [4], [9]. This again is realized in conventional MU-MIMO networks through direct signaling between clients, which is infeasible for uplink netMIMO. *Third*, it is well-known that netMIMO capacity strongly depends on the channel correlation between concurrent clients [7]. Given the large number of clients supported by an APC, it is infeasible to collect the full channel state information (CSI), especially considering the temporal variation. So, how to efficiently determine which set of clients have minimum correlation?

In this paper, we present the design and implementation of a novel cross-layer architecture, NURA (NetMIMO Uplink Random Access), that enables uplink netMIMO for distributed asynchronous clients. NURA introduces two key techniques to address the aforementioned challenges.

(i) NURA proposes medium-access signaling and semi-

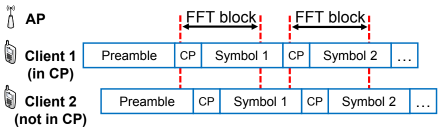

Fig. 2. Synchronization requirement in MU-MIMO (or netMIMO) OFDM uplink: receiver's FFT block operation must be aligned within all transmitters' CP.

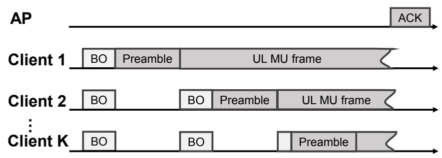

Fig. 3. Traditional media access in single-cell MU-MIMO. Each client overhears others' preambles to ensure the network's DoF has not been used up.

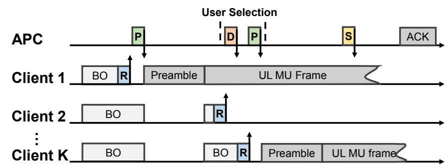

Fig. 4. NURA enables random-access based net-MIMO using a request-and-permission signaling mechanism, allowing a client to signal with the APC while other uplink transmissions are ongoing.

synchronization protocols that allow clients to randomly access the channel and send packets to the APC without interrupting ongoing uplink transmissions. Such protocols are the enabling procedure for random-access uplink netMIMO and, to our knowledge, have not been addressed in prior netMIMO or MU-MIMO research [3], [4], [7].

(ii) NURA employs a novel request-permission based user admission mechanism that allows the APC to efficiently screen those clients whose channels are correlated with ongoing clients', thereby optimizing uplink throughput.

We have implemented a NURA prototype on a netMIMO platform built from the WARP software-radios [10]. Our implementation of NURA's medium-access signaling, semi-synchronization, user admission and uplink multi-user OFDM packet decoding, verifies the feasibility of NURA's designs. By running NURA on a netMIMO testbed with 4 dAPs and 25 clients, we show that its semi-synchronous signaling mechanism can effectively coordinate the random access between distributed, oblivious clients. NURA's user admission mechanism demonstrates a throughput gain of 37% to 58% over state-of-the-art user-selection protocols [3], [4], [11], under realistic traffic patterns and node mobility.

The rest of the paper is organized as follows. Sec. III briefly introduces the architecture and challenges in uplink netMIMO. We then describe the design of NURA in Sec. IV. After describing the implementation of NURA in Sec. V, we evaluate its performance in Sec. VI. Finally, we discuss related work in Sec. II and conclude the paper in Sec. VIII.

II. Related Work

Recently, the feasibility of netMIMO has been theoretically and empirically verified. The coordinated multi-point (CoMP) architecture, proposed by 3GPP, essentially aims to push netMIMO into practice by either synchronizing existing base stations, or deploying remote antenna heads. NetMIMO is also being discussed in the next-generation WLAN standard 802.11hew [12]. Downlink netMIMO protocols are easy to handle because transmission entities (*i.e.*, APs) can be managed by a master AP [7]. Yet to our knowledge, no prior work in netMIMO has systematically addressed the uplink coordination among spatially distributed, asynchronous clients that may not even sense each other.

Maximizing user orthogonality is the most critical requirement in MU-MIMO [2]. OPUS [11] incrementally selects downlink MU-MIMO users through orthogonal probing. MIMOMate [4] allows a AP to decide user selection using historical CSI. Both schemes assume clients start transmission simultaneously only after AP determines the user grouping. Since all MU-MIMO clients fall in one cell, the signaling and coordination becomes much easier than netMIMO.

The medium access signaling mechanism in NURA is partly inspired by existing protocols that leverage energy bursts in OFDM subcarriers to piggy-back information. Via such energy-based signaling, Back2F [13] and FlashLinQ [14] reduce contention and scheduling overhead. Flashback [15] establishes a low bit-rate control plane, but requires a high transmit power and sacrifices certain data bits. NURA's medium-access signaling design evades the requirement and allows accurate detection even when the signaling SNR is lower than that of ongoing transmission.

To satisfy the uplink synchronization requirement, previous protocols [9], [16] need to use a reference frame, or a longer cyclic prefix, which incurs significant signaling overhead. Synchronization can be alternatively realized for MU-MIMO clients that can sense each other [3], but is often infeasible in NURA's netMIMO setting.

III. Background and Motivation

A. Architecture of NURA

NURA builds on a hierarchical netMIMO [7] architecture (Fig. 1). Multiple distributed APs (dAP) are connected to one master AP (mAP) via a wireline backhaul to form an AP cluster (APC). The mAP and dAPs mutually share synchronization clocks as well as data packets. Such a netMIMO APC can share a similar PHY-layer modulator/demodulator as the traditional MU-MIMO AP [3]–[5]. However, the network size of netMIMO can be multi-folds compared to a MU-MIMO cell. Clients spread over a wide area (across multiple traditional WLAN cells) and may not sense/overhear each other. Besides, the uplink SNR from one client to different dAPs can vary significantly. These characteristics differ significantly from conventional MU-MIMO protocols [3], [4], [8], [11].

NURA mainly addresses the uplink random access in a single APC. Multi-APC case will be discussed in Section VII. Each dAP/client is assuming a single antenna. Extension to a multi-antenna case entails a straightforward modification to the PHY layer, and is left out of our scope.

B. A Primer on Uplink NetMIMO PHY Layer

The APC follows two steps to decode concurrently received uplink packets: *(i) Multi-user MIMO decorrelation*, which separates individual clients' uplink data signal from overlapped signals; *(ii) Packet demodulation*, which decodes each client's data bits out of the separated single-user data signals.

1) A primer on packet demodulation: OFDM is the *de facto* PHY in most modern wireless systems, *e.g.*, 802.11a/g/n. It divides a frequency band into multiple *subcarriers*. The sender can load N data symbols into N-subcarriers, and run IFFT to convert them to an *OFDM symbol* with N time-domain samples that are sent over the air. The receiver runs FFT to recover the data symbols in frequency domain. Remarkably, this is feasibly only if the receiver synchronizes its FFT operation to the OFDM symbol from the sender.

In practice, each data packet contains a known preamble and multiple OFDM symbols. Receiver uses the preamble as a synchronization point. To tolerate minor synchronization offset between transmitter and receiver, each OFDM symbol is prepended with a *cyclic prefix* (CP) (Fig. 2). In 802.11, an OFDM symbol contains $N = 64$ samples, the 16 tailing samples being replicated as CP. Together they form an 80-sample *OFDM block*. For each packet, as long as the FFT operation starts within the OFDM block's CP, the OFDM symbol can be decoded [9].

2) MU-MIMO decorrelation for the uplink: Zero-forcing beamforming (ZFBF) is a widely used method to separate concurrent uplink packets. Consider K clients concurrently transmitting to M dAPs. Let x_i denote the symbol of client i, and vector $\mathbf{X} = [x_1, x_2, \ldots, x_K]^T$. $\mathbf{H}$, an $M \times K$ matrix, denotes the channel state information (CSI) between clients and the APC. Then, the APC's received signals can be represented by $\mathbf{Y} = \mathbf{HX}+\mathbf{N}$, where $\mathbf{N}$ is a $1\times M$ vector denoting the noise. To separate multiple signal streams, APC applies a complex matrix $\mathbf{W}$ to received signals $\mathbf{Y}$. For ZFBF, it can be computed from the pseudo inverse: $\mathbf{W} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$. The resulting decorrelated signals become $\hat{\mathbf{X}} = \mathbf{X} + \mathbf{WN}$.

The separated signal for the i^{th} client is $\hat{x_i} = x_i + \sum_{k=1}^{M} w_{ik}n_k$, where n_k is the noise at the k^{th} dAP and w_{ik} is the entry in matrix $\mathbf{W}$. Assuming the noise to be i.i.d. Gaussian with power N_0, SNR of i^{th} client's uplink is:

$$\begin{aligned} SNR_i &= \frac{E\{|x_i|^2\}}{\sum_{j=1}^{M}|w_{ij}|^2 N_0} = \frac{E\{|x_i|^2\}}{\sum_{j=1}^{M}|\sum_{k=1}^{K} b_{ik}h^*_{jk}|^2 N_0} \\ &= \frac{\sum_{l=1}^{M} E\{|h_{li}x_i|^2\}}{\sum_{l=1}^{M}\sum_{j=1}^{M}|(b_{ii}h^*_{ji}h_{li} + \sum_{k\neq i}^{K} b_{ik}h^*_{jk}h_{li}|^2 N_0} \end{aligned}$$

where b_{ik} is the element in matrix $(\mathbf{H}^H\mathbf{H})^{-1}$. Note that $b_{ii} = \frac{1}{\sum_{j=1}^{M}|h_{ji}|^2}$. If $\mathbf{h_i}^H\mathbf{h_k} = 0$ for $i \neq k$, where $\mathbf{h_i}$ denotes the i^{th} column of matrix $\mathbf{H}$, we have $b_{ik} = 0$. In this case, the SNR of the i^{th} client $\frac{\sum_{l=1}^{M} E\{|h_{li}x_i|^2\}}{N_0}$ equals to its original SNR without concurrent transmission. When $\mathbf{h_i}^H\mathbf{h_k} \neq 0$, the SNR is reduced. This is because *to eliminate mutual interference, ZFBF projects the received signal of the targeted user to a direction orthogonal to other clients*. The projected SNR is naturally lower than original SNR unless client i is perfectly orthogonal to any other client k, *i.e.*, $\mathbf{h_i}^H\mathbf{h_k} = 0$, for $i \neq k$.

Observing that $\mathbf{h_i}$ represents the CSI vector from client i to all dAPs, we can quantitatively characterize the *orthogonality* between two clients i and k as:

$$\theta_{ik} = acos(\frac{|\mathbf{h_i}^H\mathbf{h_k}|}{||\mathbf{h_i}||||\mathbf{h_k}||}). \tag{1}$$

Based on the above reasoning, we can conclude that *to maximize the sum bit-rate of all concurrent uplink transmissions, it is critical to select a group of orthogonal clients with pairwise orthogonality as close to* 90° *as possible*.

C. Challenges to Uplink NetMIMO

To enable efficient random access for uplink netMIMO, NURA must address two new challenges: *(i)* enable contention between oblivious clients who cannot directly coordinate with each other; *(ii)* ensure each newly admitted client has strong orthogonality with those who are already transmitting.

1) Coordinating Random Access in Uplink NetMIMO: In uplink netMIMO, though APC can assign transmit opportunities to clients without contention, DoFs allocated to inactive clients will be wasted, since the APC has no prior knowledge of whether or which clients have packets queued up. The random access netMIMO mechanism of NURA naturally circumvents the issue.

Realizing the mechanism entails two key problems: *(i) Clients should contend for uplink access without carrier-sensing/overhearing*. Ideally, APC can execute a handshake with a newly joining client and inform it whether there exists extra DoF and whether it is allowed to transmit. However, such signaling procedure can easily interfere with ongoing uplink transmissions. *(ii)* Although the starting time of different transmissions are not required be simultaneous, *the OFDM symbol boundary of a new uplink transmission must be aligned within CP to existing transmissions*. Existing protocols satisfy this requirement by allowing AP to broadcast a reference frame [9], so that clients can start transmission synchronously. But this breaks the random access principle of NURA.

2) Tailoring User Admission for NetMIMO: Selecting orthogonal users is a common theme in conventional MU-MIMO networks [4], [11]. In OPUS [11], for example, a MU-MIMO AP can broadcast a probing frame, and a new client is admitted to join only if it has strong orthogonality with existing clients. However, the probing frame assumes the AP has co-located antennas, such that each client senses similar RSS from them. This no long holds true in netMIMO with widely distributed dAP antennas. Alternatively, a MU-MIMO AP can collect CSI from all clients, centrally compute and assign the group with maximum orthogonality, as proposed in MIMOMate [4]. However, in a netMIMO cluster, collecting and updating CSI from all clients will incur formidable overhead considering the large user base and channel variations.

Therefore, a new client admission/selection scheme has to be designed for the uplink of netMIMO WLANs. A client should win channel access only if it has strong orthogonality to clients who are already transmitting. Orthogonality needs to be gauged at low overhead so as to be applied to a large network and rapidly changing environment.

IV. NURA Design

A. Overview of Protocol Operations

NURA is a cross-layer design to meet the aforementioned challenges. Simply put, a NURA APC runs a lightweight *medium access signaling* (MAS) scheme to arbitrate clients' random access. The APC uses a *user-admission mechanism* to gauge a new user's orthogonality to existing ones, without extensive probing overhead. Meanwhile, clients adopt a

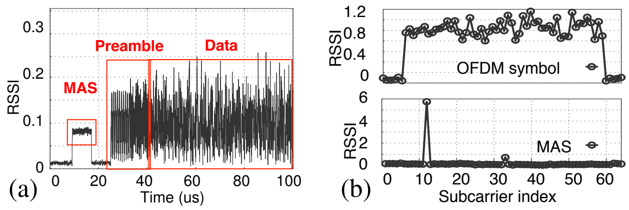

Fig. 5. (a) MAS and data packet in time domain. (b) Received energy burst concentrates on the correct subcarrier despite sync timing offset.

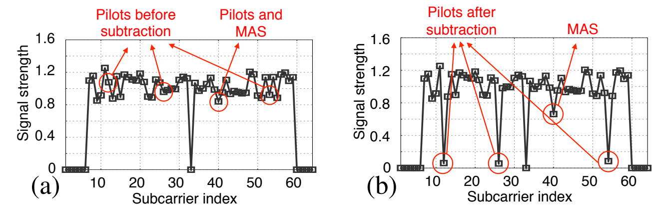

Fig. 6. Remove pilot symbols by subtraction of adjacent OFDM blocks to enhance detection robustness. (a) before and (b) after removal.

semi-synchronization mechanism to satisfy the OFDM symbol alignment. All three signaling primitives must satisfy one challenging requirement: they should not affect uplink transmissions that are already going on.

Fig. 4 illustrates a typical random access procedure in NURA. Clients contend with a random backoff protocol similar to 802.11, but instead of carrier sensing, they rely on a request-and-permission signaling with the APC (which may be in the process of receiving another client's uplink packet). More specifically, the clients and APC interact as follows:

(i) When a client finishes random backoff (BO), it sends a *request* (R) signal. Upon detecting the request, the APC employs the *user admission* mechanism to estimate the client's orthogonality *w.r.t.* ongoing clients, and sends a *permission* (P), or *deny* (D) signal through the dAP who detected the strongest request signal. All the signals are sent via the *MAS* scheme, which ensures no interference to ongoing uplink transmissions.

(ii) If a client obtains a P, it will align to ongoing transmissions' OFDM symbol boundary via the *semi-synchronization* mechanism, and then start the data transmission. When overhearing a P, irrelevant clients in backoff state (*e.g.*, client 2 in Fig. 4) will temporarily freeze for 2 time slots ($18\mu s$) to avoid starting too early and corrupting the preamble of the newly admitted client.

(iii) When its DoF is used up, APC disables new requests via a *stop* (S) signal. Upon decoding the uplink packets, the APC simultaneously sends multiple ACKs via downlink netMIMO as in [7].

A cycle from step (i) to (iii) is called *one round of transmission*. Afterwards, the APC may start a new round or switch to downlink transmission.

We now proceed to detail the design components in NURA.

B. Media Access Signaling (MAS)

MAS is the request-and-reply mechanism between a newly joining client and the APC, which includes 4 types of signals mentioned above: request, permission, deny and stop. The key design objective is to send the signals in a covert manner *before the new client is synchronized to the APC, and without disturbing ongoing uplink transmissions*.

1) Basic MAS: To meet the objective, NURA generates MAS as an energy burst concentrated on a particular subcarrier in an OFDM block. The receiver needs to correctly detect the energy burst on that subcarrier without relying on any preambles (which will otherwise disturb ongoing transmissions).

To create an MAS, the transmitter puts "1" on the intended subcarrier and "0" on the others in an OFDM symbol; and then converts it to a time-domain symbol by IFFT. The symbol is duplicated by two and a half times, equivalent to 2 OFDM block duration (top of Fig. 5(a)). The resulting MAS is a cyclic sequence in time, and any FFT block taken within it will have the same energy concentrating on the intended subcarrier (proof in Proposition 2). So, the receiver can detect the MAS via its OFDM decoder without symbol-level synchronization.

We verify the above design by sending an MAS followed by a data packet in our testbed (Sec. V). Fig. 5(a) shows the snapshot of received signals. Fig. 5(b) plots the received OFDM symbols and MAS mapped in frequency domain. Although the transmitter is not synchronized with the receiver, the energy of received MAS still concentrates on the correct subcarrier with marginal energy leakage to adjacent subcarriers.

2) Detecting MAS Mixed with Data: The basic design assumes there is no ongoing transmission. But in practical netMIMO, NURA needs to work even when MAS is mixed with ongoing data transmission, and even when the SNR of MAS signal is lower than that of ongoing transmission.

NURA addresses the problem using a *pilot subtraction* scheme. An OFDM block contains 4 known pilot subcarriers to compensate the carrier frequency offset (CFO). Pilot symbol in a subcarrier, subject to the same channel distortion, remains similar across adjacent OFDM blocks. Therefore, the receiver can nullify the pilot signals in ongoing transmissions by subtracting pilot subcarriers of neighboring OFDM blocks, leaving only the MAS energy burst. Fig. 6 shows a measured OFDM block before and after subtraction, where the MAS is clearly separated. NURA's 4 types of MAS signals each is sent via one pilot subcarrier.

Pilot subtraction only sacrifices one OFDM symbol's pilots, originally used for CFO estimation/compensation. A packet contains hundreds of OFDM symbols, and CFO is usually stable over minutes to hours [9]. Hence, it suffices to reuse pilots from other adjacent OFDM symbols for CFO estimation.

MAS requires the APC to serve ongoing uplink transmissions while occasionally replying energy bursts to new clients. This can be done through two separate dAPs. The APC does *not* need to recover the received pilot symbols that are corrupted by its own reply signals. Thus, *it does not require a sophisticated full-duplex radio* [17] to eliminate the impact of self-interference. In addition, NURA only requires the dAP closest to the newly joining client to send the reply signal, which takes advantage of the unique topological structure of netMIMO and further reduces the impact on ongoing uplink transmissions close to other dAPs.

3) Collision Analysis: In NURA, MAS collision happens when a client sends MAS concurrently with a dAP or another

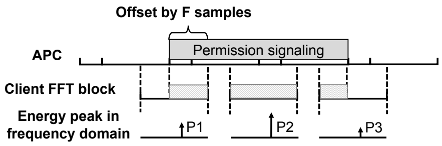

Fig. 7. APC-assist synchronization: finding the timing offset by examining the MAS signal strength difference between adjacent FFT blocks.

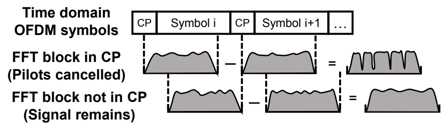

Fig. 8. Client-assist synchronization mechanism: finding the timing offset by locating the FFT block which can remove the pilot signal.

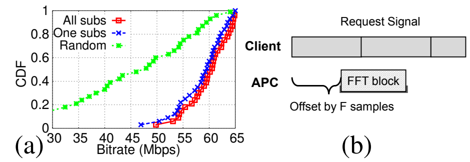

Fig. 9. (a) MAS based single-subcarrier selection achieves bitrate (96%) close to the all-subcarrier case, and 1.34× better than random selection. (b) Illustration of data timing offset of request signaling.

client. For the first case, since the dAP and client's MAS concentrates on different subcarriers, they can still be detected even when overlapping. For the second, recall that MAS lasts for two 802.11 OFDM blocks ($8\mu s$). The duration of a backoff time slot is $9\mu s$. Unless two clients have exactly the same backoff counter and start contention simultaneously, they are unlikely to collide. Thus, for clients around each dAP, the collision probability will be similar to that in 802.11 networks.

In the rare case when collision occurs, how will it affect the system? The APC may proceed to reject or accept collided clients. The former case is trivial. In the latter case, the collided clients' data cannot be decoded. But because they are orthogonal to other concurrent clients (controlled by the admission policy which will be detailed in Section IV-D), the collision only affects themselves [7].

C. Semi-synchronization Mechanism

Recall in NURA, although the clients randomly access the channel, the starting point of their transmission must be aligned within the CP of one OFDM block in ongoing uplink transmissions (Sec. III-C1). The CP-level alignment is referred as *semi-synchronization*. NURA designs two complementary mechanisms – APC-assist and client-assist synchronization to realize semi-synchronization. They are designated for newly joining clients who overhear different levels of signal strengths (denoted as Φ_I) from ongoing transmissions.

1) APC-assist Synchronization: When Φ_I is small, client can opportunistically overhear the permission signal from the APC to achieve semi-synchronization, as shown in Fig. 7. To synchronize to the APC, the newly joining client runs 3 consecutive FFT blocks that (partially) overlap with the received permission signal. Suppose the permission signal amplitudes over these three FFT blocks are P_1, P_2, and P_3 respectively, then the client leverages the following proposition to compute the timing offset F *w.r.t.* the APC.

Proposition 1 *The signal strength of permission signal in an FFT block is proportional to the number of samples of permission signal in that FFT block. Supposing FFT block has N points, the corresponding timing offset estimated from the first and second FFT blocks is $F_{12} = NP_1/P_2$.*

Proof: Available in the technical report [18]. □

In a similar way, timing offset can be estimated from the second and third FFT blocks as $F_{23} = \frac{N(P_2-P_3)}{P_2}$. Combining both results, the estimated timing offset $F = \frac{F_{12}+F_{23}}{2} = \frac{N(P_1+P_2-P_3)}{2P_2}$. When there exist ongoing transmissions, the estimation error of the approach can be approximated by: $\xi \approx \frac{\Phi_I N}{2(P_2-\Phi_I)}$ (see [18] for details). In 802.11 OFDM, $N = 64$. Suppose the maximum tolerable timing offset $\xi = 4$ samples, then $\Phi_I/P_2 = \frac{1}{9}$. The maximum power ratio between ongoing transmission and permission signal overheard by the new client should thus satisfy $20\log_{10}(\frac{\Phi_I}{P_2}) < -19$ dB.

2) Client-assist Synchronization: When a following user misses the permission signal to the first client and senses unusually strong signal Φ_I from active peers, it can leverage that signal to synchronize, referred to as *client-assist synchronization*. The underlying assumption is that the active peer's transmission must have already aligned with the APC through one of synchronization mechanisms. The question is how does a client leverage that signal for synchronization, considering that there may be multiple overlapping transmissions?

Our solution resides in an observation illustrated in Fig. 8. If the new client's FFT operation aligns with the OFDM block of ongoing transmission, subtracting two adjacent FFT blocks with each other can remove the pilot symbols, creating dramatic notches in the frequency domain. Otherwise, if the FFT operation spans over two OFDM blocks, the FFTed signals are random, and pilot signals cannot be removed.

Therefore, the client can slide its FFT block and locate the timing offset when it observes the notches due to pilot subtraction. To detect the notches, it uses the signal strength of data symbols (from ongoing transmissions) as reference, which reflects the strength of pilot symbols before subtraction. If the energy ratios of 4 candidate notches to the data symbol $\frac{E_{pilot}}{E_{data}}$ are all less than a threshold, which is empirically set as 0.4, the client is considered to be synchronized within the CP.

D. Orthogonal Client Admission

NURA's orthogonal user admission mechanism allows APC to grant permission to a new client only if it has good pairwise orthogonality with ongoing uplink transmitters. To compute the orthogonality (Eq. (1)), the APC needs CSI from clients. But before granting permission to a client, the APC does not have the client's preamble and cannot estimate its channel.

To resolve the dilemma, the APC estimates the orthogonality of the new client using the CSI estimated from the client's MAS request signal. This design begs two questions: *(i)* Can the energy-burst based MAS request signal be harnessed to evaluate orthogonality between users? *(ii)* The client has not been synced to APC when sending requests. So how to extract the orthogonality estimation and how does the lack of synchronization affect the result?

Orthogonality Approximation. To answer the first question, we emphasize that orthogonality differs from frequency diversity. The former depends on the relative channel angle between two users (see Eq. (1)), whereas the latter depends on the multipath fading of each individual user. Even in environment with high frequency diversity, two users may have consistently strong (or weak) orthogonality across all subcarriers. In addition, when evaluating the orthogonality between two users, one has to rely on a metric that reflects the average-case orthogonality across all subcarriers. Otherwise, one would need user selection on a per-subcarrier basis, which exacerbates the signaling overhead (*e.g.*, by 48× given 48 data subcarriers in 802.11). Therefore, in NURA, we insist on a lightweight approach: using the CSI estimated from the short MAS signal to approximate the average-case orthogonality.

We corroborate the orthogonality approximation through an empirical study. We conduct experiments on a software-radio testbed with a netMIMO OFDM PHY implementation (Sec. V). Without loss of generality, we run an APC containing 2 dAPs and 30 clients randomly distributed over different positions. Each time a set of 5 clients will be randomly selected. We collect their CSIs, exhaustively search the optimal groupings by using one random subcarrier (used by MAS) and all subcarriers respectively. We obtain the bit-rate of uplink by computing the decoded symbols' SNR and mapping it to bit-rate following [7]. Experiments run in a 29×36ft lab environment with metal cabinets, tables, partitions and drywalls, which have rich frequency diversity across subcarriers.

Fig. 9(a) shows the CDF of uplink bit-rate distributed across all possible user groups. MAS based user selection only experiences minor bit-rate loss (<4%) compared to the all-subcarrier case, and has substantial gain over random user selection. This micro-benchmark test implies that, despite its suboptimality, orthogonality obtained through MAS can well approximate the average-case performance of an exhaustive approach using all subcarriers. More system-level tests and further verifications will be conducted in Sec. VI.

Orthogonality Estimation. To answer the second question, we first make the following observation.

Proposition 2 *The lack of synchronization of the MAS request signal will cause an unknown but constant phase shift of the CSI estimation on a subcarrier. This constant phase shift does not affect the orthogonality relation between clients.*

Proof: Let X_k and x_n denote the request signal in frequency domain and time domain respectively. Suppose APC runs an N-point OFDM block, by the Fourier transform definition, we have $X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}}$. At the APC side, as illustrated in Fig. 9(b), the FFT block will span two blocks of request signaling. Let $\hat{X}_k$ and $\hat{x}_n$ correspond to received signals in frequency and time domain, and assume the timing offset of the request signaling is F samples. We can readily have following relations: $\hat{x}_n = x_{n+F}$ for $0 \le n \le N-F-1$ and $\hat{x}_n = x_{n-(N-F)}$ for $N-F \le n \le N-1$. By applying FFT to $\hat{x}_n$, we have

$$\begin{aligned}\hat{X}_k &= \sum_{n=0}^{N\text{-}1} \hat{x}_n e^{\frac{-j2\pi kn}{N}} = \sum_{n=0}^{N\text{-}F\text{-}1} x_{n+F} e^{\frac{-j2\pi kn}{N}} + \sum_{n=N\text{-}F}^{N\text{-}1} x_{n\text{-}(N\text{-}F)} e^{\frac{-j2\pi kn}{N}} \\ &= \sum_{n=F}^{N-1} x_n e^{\frac{-j2\pi k(n-F)}{N}} + \sum_{n=0}^{F-1} x_n e^{\frac{-j2\pi k(n+(N-F))}{N}} \\ &= e^{\frac{j2\pi kF}{N}} \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi kn}{N}} = X_k e^{\frac{j2\pi kF}{N}}.\end{aligned}$$

The request signal is sent over a single subcarrier. Let $h_{ij,d}$ denote the channel between client j and dAP i, and d represent the index of subcarrier. Then the received symbols at all dAPs become:

$$\begin{aligned}\hat{\mathbf{X}}_\mathbf{j} &= [h_{1j,d} X_d e^{\frac{j2\pi dF}{N}}, \ldots, h_{ij,d} X_d e^{\frac{j2\pi dF}{N}, \cdots}]^T \\ &= \mathbf{h_j} X_d e^{\frac{j2\pi dF}{N}}.\end{aligned}$$

By orthogonality in Eq. 1, we have $\frac{|\hat{\mathbf{X}}_\mathbf{j}^H \hat{\mathbf{X}}_\mathbf{k}|}{||\hat{\mathbf{X}}_\mathbf{j}||||\hat{\mathbf{X}}_\mathbf{k}||} = \frac{|\mathbf{h_j}^H \mathbf{h_k}|}{||\mathbf{h_j}||||\mathbf{h_k}||}$, which means the received request signal can be leveraged to estimate the orthogonality of clients, thus completing the proof. □

Client Admission. The orthogonal client admission scheme allows an APC to leverage the above observations to admit/reject client request. It ensures each newly admitted client has sufficient orthogonality with ongoing clients, and hence high potential to boost uplink capacity.

More specifically, the APC always admits the *first* client that wins random access in a transmission round. For the second client and so on, the APC runs an admission policy. Let $r(u_1, \ldots, u_j)$ denote the sum bit-rate for client grouping $(u_1, \ldots, u_j)$. To estimate $r(\cdot)$, APC computes and maps the projected SNR (Sec. III-B) to achievable bit-rate (Sec. V). When the m^{th} client attempts to join, APC admits it only if:

1) $r(u_1, \ldots, u_m) > \beta^m \sum_{i=1}^m r(u_i)$
2) $r(u_1, \ldots, u_m) > r(u_1, \ldots, u_{m-1})$

The first criterion implies the sum rate of admitted clients should be larger than that in single-user transmission discounted by a factor β^m. The rationale is to control the strictness of orthogonality requirement for the admitted clients. When $\beta = 1$, all clients should be pairwise-orthogonal. When $\beta < 1$, certain imperfection can be tolerated. The exponent m accounts for the channel hardening effect [19], *i.e.*, in practice the throughput cannot grow linearly – the improvement diminishes as more concurrent clients are added, because it becomes harder to find a client that is orthogonal to all existing ones as m grows.

The second criterion indicates the sum throughput should not decrease after admitting the new clients. It complements the first one in order to prevent the channel hardening effect from lowering the system's performance.

V. Implementation and Experimental Setup

A. Implementation

We have built a prototype of NURA using WARP [10], a fully programmable software-radio platform. We first implemented an OFDM PHY-layer library, with similar modules and time/frequency parameters as 802.11g 20MHz mode. One notable aspect is that due to widely distributed dAPs, not all of them can detect each uplink packet. Fortunately, since the dAPs are synchronized, we simply leverage the one with strongest signal to perform preamble detection and synchronization. On top of the OFDM PHY module, we implement the ZFBF-based netMIMO decoder (Sec. III-B), MAS generator/detector (Sec. IV-B), semi-synchronizer (Sec. IV-C) and orthogonal client selector (Sec. IV-D).

Our NURA prototype runs on a netMIMO testbed consisting of one APC and multiple clients. Two original WARP

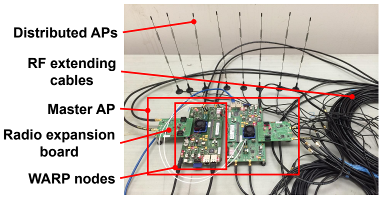

Fig. 10. NURA prototype using WARP.

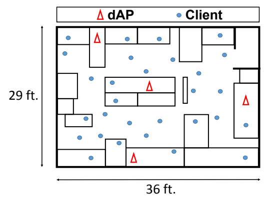

Fig. 11. Testbed topology.

boards (each with a FMC-RF-2X245 module) are carrier-synchronized to act as an mAP. We extend 4 dAPs from the mAP using 25-ft LMR-240 RF extension cables (Fig. 10). Each dAP has separated Tx/Rx chains to receive packets and transmit MAS simultaneously. 4 other WARP boards are used as single-antenna clients, and moved to different locations to create diverse topologies. Since the RF cables have 10dB loss, they act as attenuators that reduce the clients' transmission range to around 30ft at 100 mW transmit power. The reduced dAP coverage allows us to emulate a large netMIMO cluster in a relatively small office environment (Fig. 11).

B. Experimental Setup

Our implementation realizes real-time MAS generation and detection. Due to limited hardware, for design components that involve more than 4 clients, we collect and process trace data offline. Specifically, each client at a certain location transmits 200 1.5-KB packets, whereas all dAPs store the received signals without decoding. As wireless signals are linearly superposable in time-domain, to emulate the request, semi-synchronization and concurrent transmission procedures, we process, segment, align and sum the raw signals of clients for each dAP, and then feed the resulting signals to the uplink netMIMO decoder. By default, the uplink data are QPSK modulated. We measure the SNR of decoded symbols, and then map it to achievable bit-rate similarly to [7].

Note that a new client's preamble may be overlapped with ongoing data transmissions, but can be restored using *successive interference cancellation (SIC)* [3]–[5]. The overlapped data can be first decoded through a subset of dAPs minimally interfered by the new client. Then the APC re-encodes the data and subtracts them from received signals, thus restoring the new client's preamble and CSI. Now that the APC has CSI of all concurrent clients, it can proceed with the ZFBF decoding (Sec. III-B). For simplicity, our implementation restores the overlapped preamble by subtraction using raw signals to emulate the result of SIC.

For benchmark comparison, we are unaware of any other uplink netMIMO protocols for WLANs. So we instead improve existing MU-MIMO co-located antenna systems, including MIMOMate [4], OPUS [11] and SAM [3] (Sec. III-C), but faithfully retain their user selection schemes. They explicitly separate each round of transmission into user selection and concurrent transmission stages. In MIMOMate, clients randomly contend for the first DoF, and AP explicitly assigns the remaining DoFs to other clients via optimal matching. To fit MIMOMate to netMIMO, we add RTS/CTS exchange to resolve invisibility between clients. When a client's backoff counter first reaches zero, it sends an RTS to the APC. Sequent assignment of remaining DoFs will be specified and broadcast by the APC in a single CTS packet. Active client confirms its assignment by replying another RTS. Unused DoFs of inactive clients will be re-contended by others. We call this improved protocol MIMOMate+. OPUS and SAM are modified in a similar way, referred to as OPUS+ and SAM+.

Since WARP does not support real-time MAC [11], we build a MAC emulator on top of our PHY implementation, to account for the overhead and throughput for all four schemes. The emulator uses a virtual clock to execute protocol operations including random backoff, DIFS/SIFS, RTS/CTS, MAS, ACK, *etc.*. Whenever allowed by the protocol, it starts uplink netMIMO transmission, decodes the data to obtain the bit-rate and packet duration accordingly.

VI. Evaluation Result

A. Micro-benchmark Evaluation

1) MAS Detection Accuracy: The key factor that determines the MAS detection accuracy is ΔS, defined as the dAP-received ongoing transmission SNR minus MAS SNR. To evaluate the accuracy, we let one client send data packets while the other transmits MAS to the same dAP – this creates the worst-case interference because the impact will be much smaller when two clients are close to different dAPs. The results in Fig. 12 show that more than 99% MAS can be detected if $\Delta S < 10$dB. The false negative becomes noticeable when $\Delta S > 10$dB because the residual noise increases due to imperfect pilot subtraction. But owing to the widely distributed dAPs, as long as one dAP experiences a small ΔS, the MAS can be detected just by that dAP. Fig. 13 shows the CDF of minimum ΔS among all dAPs in the test topology. Almost all ΔS values fall below 0dB, corresponding to close to 0 false-negative/positive in Fig. 12. Therefore, MAS can be detected with high accuracy in realistic netMIMO scenarios.

2) Does MAS Affect Data Transmission?: To answer this question, we conduct an experiment with similar settings as above. Fig. 14 shows that the impact is less than 2% when fewer than 12 MAS ($\Delta S = -10dB$) are sent across one packet. The loss comes from imperfect CFO estimation and small leakage of energy burst. In a netMIMO with 4 dAPs and 25 clients, we observe median of 14 requests over one round of transmission (Fig. 15). Note that only the client who started first will experience this amount of overhead. Even in such case, the impact on throughput is only 8%. Fig. 14 further shows that as long as ΔS is not too small (*i.e.*, $\Delta S > -12$dB), the ongoing transmission's bitrate is unaffected by MAS. Such requirement can be easily satisfied as we already observed in Fig. 13 that most of case the minimum ΔS is greater than -10dB.

3) Accuracy of Semi-synchronization: Recall either APC-assist or client-assist scheme can be used, depending on the SNR of ongoing transmission overheard by the newly joining client. Fig. 16(a) and 16(b) plot the CDF of timing offset after running the semi-synchronization scheme, under different ranges of overheard SNR. Since the CP length equals 16, the client achieves synchronization if its timing offset falls within 0$\sim$15 samples. We can see that although the sync accuracy

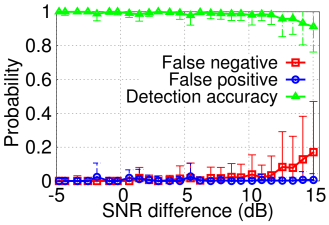

Fig. 12. MAS detection accuracy over ΔS.

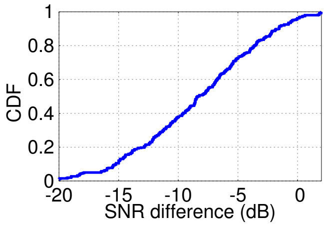

Fig. 13. CDF of minimum ΔS *w.r.t.* to all dAPs in a netMIMO cluster.

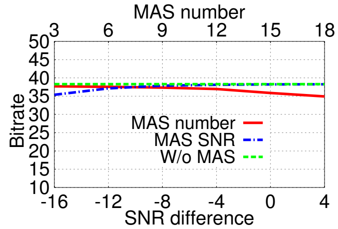

Fig. 14. Impact of MAS on bitrate.

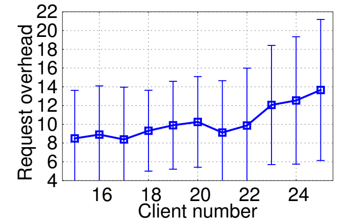

Fig. 15. Request overhead.

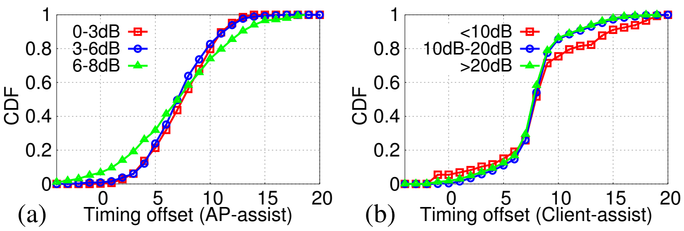

Fig. 16. Accuracy of semi-synchronization mechanism in NURA.

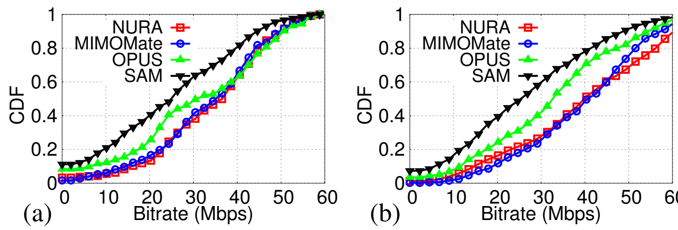

Fig. 17. Achievable bitrate under different user selection schemes. Client number = 15. (a) 2 dAPs. (b) 4 dAPs.

is relatively lower in the transition range of the two methods, *i.e.* 6$\sim$10dB overheard SNR, even in such worst case, the sync requirement is satisfied with $> 90\%$ confidence. Overall, 95% of test cases satisfy the requirement.

4) Efficacy of Orthogonal User Admission: Fig. 17 (a) and (b) plot the CDF of bitrates for different user selection schemes, distributed across all 15 clients. NURA achieves comparable bitrate $(0.93\times)$ with MIMOMate that centrally collects CSI and computes the optimal user group. We will demonstrate later that NURA's advantages manifest under client mobility and non-saturated traffic demands. Besides, NURA achieves bitrate gain 1.21–$1.24\times$ and 1.47–$1.53\times$ over OPUS and SAM. OPUS's probing-based mechanism no longer works in a netMIMO topology (Sec. III-C), whereas SAM leads to the lowest bit-rate because of lack of user selection.

5) Fairness: We evaluate the fairness of media access opportunity under different client population and topologies with 4 dAPs. Fig. 18 plots the Jain's fairness index of transmit opportunities across 300 rounds of concurrent uplink transmissions. We can see that NURA's fairness ranges from 0.83 to 0.92 and is relatively stable over client number. Its fairness also fluctuates negligibly in different topologies (Fig. 19), each with 15 randomly selected clients in our testbed. SAM is media access fair because clients are randomly selected. MIMOMate centrally selects users under fairness constraint. Thus, it achieves similar fairness as SAM. For OPUS, clients with imbalance received signals are less likely to be selected because it fails to differentiate their orthogonality (Sec. III-C), and thus has relatively poor fairness.

6) MAC and Computation Overhead: Fig. 20 evaluates the overall throughput taking into account all MAC overhead with 4 dAPs and 15 clients. In contrast to Fig. 17(b), NURA is now $1.12\times$ of MIMOMate+ owing to lightweight signaling. OPUS+ incurs larger overhead and lower throughput due to its binary-count-down contention. Fig. 21 evaluates the computation time for each user-selection scheme in a transmission round. Although the computation runs on a commodity PC, the results roughly reflect the relative computational cost. From 15 to 25 clients, MIMOMate requires 56–$111\times$ and 103–$267\times$ computation over NURA and OPUS. Besides, the computation overhead of NURA and OPUS grows linearly while MIMOMate increases squarely with client number. The result is consistent with the analysis in MIMOMate [4].

B. System-level Test

1) Non-saturated Client: We create non-saturated traffic patterns by allowing each client to generate packets via a Poisson process with packet arrival rate λ per transmission round. When $\lambda = 1$, the average numbers of active clients in 15-client and 25-client topologies are 10.3 and 16.2 respectively. Thus, $\lambda \geq 1$ suffices to ensure active clients can saturate the DoF with high probability, although not all clients always have uplink packets. Fig. 22(a) shows that NURA achieves an average throughput gain of 1.31–$1.52\times$ and 1.35–$1.81\times$ over MIMOMate+ and OPUS+ when client number increases from 15 to 25. The lower performance of MIMOMate+ in this case mainly comes from extra overhead and orthogonality degradation when the AP assigns transmission opportunity to clients with no pending traffic (Sec. III-C). Though we find the gap between NURA and MIMOMate+ decreases as the network becomes more and more saturated (Fig. 22(b)), even in close-to-saturated case ($\lambda = 2.6$ corresponding to 23.3 active clients), NURA still outperforms it by $1.13\times$.

2) Mobility: We now evaluate the MAC throughput of a 25-client network under environment mobility (people walking by and within the topology area). The experiment runs over 500 continuous packets. Fig. 23(a) shows that NURA achieves a throughput gain of $1.37\times$, $1.52\times$ and $1.58\times$ over MIMOMate+, OPUS+ and SAM+. MIMOMate only updates the CSI of selected clients after each round of transmission. Thus, CSI of other clients who are not selected and updated timely will be outdated due to channel variation. We further evaluate the protocol for indoor mobile clients. Since the minimum packet interval on WARP is $27\times$ larger than that on wireless card, we slowly move nodes $(5-10cm/s)$ to emulate normal motion. Fig. 23(b) shows a similar throughput gain of NURA as the environment mobility case.

VII. Discussion

Uplink and Downlink Coexistence. NURA only addresses coordinating uplink transmissions. To coexist with a downlink

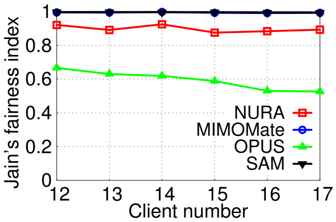

Fig. 18. Media access fairness of various client numbers.

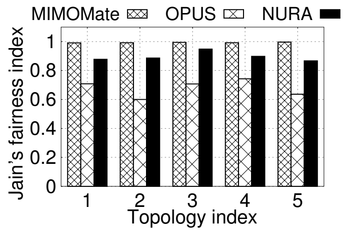

Fig. 19. Media access fairness of various topologies.

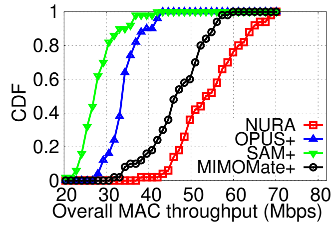

Fig. 20. MAC-layer throughput under saturated-packet arrival scenario.

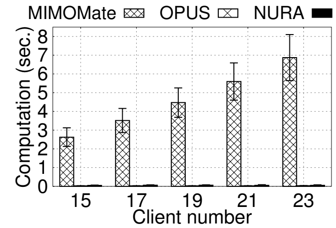

Fig. 21. Computation overhead of compared schemes.

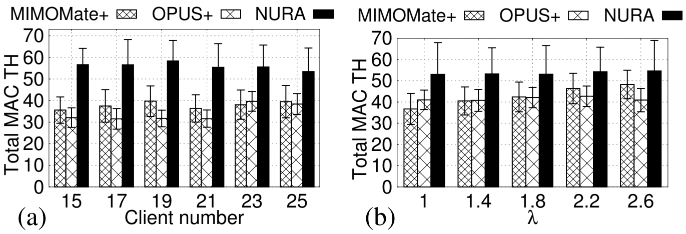

Fig. 22. Overall MAC-layer throughput for spontaneous clients with 4 dAPs. (a) Packet arrival rate $\lambda = 1$ (b) Client number = 25.

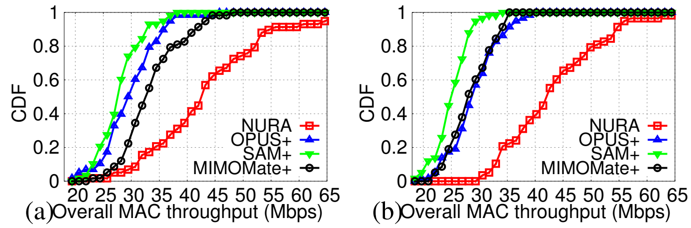

Fig. 23. Overall MAC-layer throughput under (a) environmental mobility and (b) client mobility scenarios. $\lambda = 1$

netMIMO protocol [7], we can simply reuse the phased transmission as in other multi-user MAC protocols [9], which toggles the contention priority between uplink and downlink transmissions, so that they can occur in an alternate manner.

Multiple NetMIMO Clusters. Although random access itself is amenable for distributed self-organized networks, extending NURA to multi-cluster netMIMO imposes additional challenges. For example, the MAS signals from one client may reach multiple nearby APCs. One potential solution is to employ a soft-association principle as in [7] – an APC can grant permission as long as the request has sufficient orthogonality to its existing uplink transmissions. A systematic study of such challenges and solutions is left for our future work.

Rate Adaptation. After obtaining a permission, a client needs to choose an appropriate uplink bit-rate according to its channel quality. Rate adaptation problem for MU-MIMO with co-located antenna has been addressed by [5]. It assumes clients can broadcast and mutually overhear the orthogonality and SNR information, which however cannot be directly applied to netMIMO. One heuristic is that the APC can estimate the appropriate rate through the MAS request signal and piggybacks rate information in its permission signal, but modulating the signal strength similar to Flashback [15]. To account for frequency selectivity across subcarriers, the client can send MAS through different pilots spanning an entire band, and across multiple OFDM symbols. Detailed exploration of such solutions is left for future work.

VIII. Conclusion

Enabling uplink random access has been a major obstacle for deploying netMIMO in practice. The proposed system, NURA, marks a first step to meet this challenge, enabling contention and selection of widely distributed clients that may not even sense each other. We have verified NURA through a comprehensive implementation of its MAC-level signaling and PHY-layer multi-user decoding modules. Our immediate next step is to integrate NURA with existing downlink netMIMO protocols (*e.g.*, NEMOx [7]), and extend it to large-scale multi-APC scenarios.

References

[1] N. Ericsson, "Ericsson Mobility Report, On the pulse of the Networked Society," 2012.

[2] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[3] 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.

[4] T.-W. Kuo, K.-C. Lee, K.-J. Lin, and M.-J. Tsai, "Leader-Contention-Based User Matching for 802.11 Multiuser MIMO Networks," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, 2014.

[5] 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.

[6] 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 JSAC*, vol. 28, no. 9, December 2010.

[7] X. Zhang, K. Sundaresan, M. A. A. Khojastepour, S. Rangarajan, and K. G. Shin, "NEMOx: scalable network MIMO for wireless networks," in *Proc. of ACM MobiCom*, 2013.

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

[9] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, J. Zhang, and Y. Zhang, "Fine-grained channel access in wireless LAN," *ACM SIGCOMM*, 2011.

[10] Rice University, "Wireless Open-Access Research Platform," http://warp.rice.edu/trac/wiki, 2013.

[11] X. Xie and X. Zhang, "Scalable User Selection for MU-MIMO Networks," in *Proc. of IEEE INFOCOM*, 2014.

[12] "IEEE802.11 HEW," http://www.ieee802.org/11/Reports/hew_update.htm.

[13] S. Sen, R. Roy Choudhury, and S. Nelakuditi, "No time to countdown: Migrating backoff to the frequency domain," in *ACM MobiCom*, 2011.

[14] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," *IEEE/ACM Transactions on Networking*, 2013.

[15] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath, "Flashback: Decoupled lightweight wireless control," *ACM SIGCOMM*, 2012.

[16] Y. J. A. Zhang, "Multi-round contention in wireless LANs with multi-packet reception," *IEEE Trans. on Wireless Communications*, 2010.

[17] D. Bharadia, E. McMilin, and S. Katti, "Full duplex radios," in *Proc. of ACM SIGCOMM*, 2013.

[18] T. Wei and X. Zhang, "Random Access Signaling for Network MIMO Uplink," *http://twei.ece.wisc.edu/papers/NURA-techreport.pdf*, 2016.

[19] B. M. Hochwald, T. L. Marzetta, and V. Tarokh, "Multiple-antenna channel hardening and its implications for rate feedback and scheduling," *IEEE Transactions on Information Theory*, 2004.