

Uplink MU-MIMO in Asynchronous Wireless LANs

Huacheng Zeng
University of Louisville
Louisville, Kentucky
huacheng.zeng@louisville.edu

Hongxiang Li
University of Louisville
Louisville, Kentucky
h.li@louisville.edu

Qiben Yan
University of Nebraska-Lincoln
Lincoln, Nebraska
yan@unl.edu

ABSTRACT

In wireless LANs (WLANs), network-wide time and frequency synchronization among user devices is widely regarded as a necessity for uplink MU-MIMO. Therefore, to enable uplink MU-MIMO in 802.11ax, dedicated MAC protocols (e.g., trigger frame and timing advance mechanism) have been proposed to synchronize user devices in the time and frequency domains. Such MAC protocols increase not only network complexity but also communication overhead. In this paper, we show that the time and frequency synchronization among user devices is not a necessity for uplink MU-MIMO. We propose a practical uplink MU-MIMO solution which does not require time and frequency alignments of the signals from user devices. The key component in our solution is a new PHY design for AP's receiver, which can decode the concurrent signals from multiple asynchronous user devices. We have built a prototype of our uplink MU-MIMO solution on USRP2-GNURadio testbed. Experimental results show that, using the new PHY, an M -antenna AP can successfully decode the concurrent signals from M asynchronous user devices ($2 \leq M \leq 4$).

CCS CONCEPTS

• **Networks** → **Wireless access points, base stations and infrastructure; Wireless local area networks;**

KEYWORDS

Wireless networking, uplink MU-MIMO, synchronization, multi-user detection

ACM Reference Format:

Huacheng Zeng, Hongxiang Li, and Qiben Yan. 2018. Uplink MU-MIMO in Asynchronous Wireless LANs. In *Mobihoc '18: The Nineteenth International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3209582.3209585>

1 INTRODUCTION

With the advancement of MIMO techniques, many commercial off-the-shelf (COTS) Wi-Fi routers (e.g., ASUS AC5300 and Linksys EA9500) have been equipped with eight antennas and the number of antennas on access point (AP) is steadily increasing. In contrast to

powerful AP, most of user devices (e.g., smartphones, laptops, and tablets) have only one or two antennas due to their size limit and power consumption constraint. To address the antenna asymmetry issue and fully utilize AP's spatial degrees-of-freedom provided by its antennas, multi-user MIMO (MU-MIMO) has attracted a large amount of research effects towards enabling AP to serve multiple users concurrently in both uplink and downlink.

In wireless networks, network-wide time and frequency synchronization among user devices is widely considered as a necessity for uplink MU-MIMO. With perfect time and frequency synchronization, uplink MU-MIMO is exactly the same as point-to-point MIMO (or single-user MIMO) and, therefore, conventional multi-user detection algorithms such as MMSE and zero-forcing (ZF) can be used to decode the concurrent signals at the AP. However, achieving time and frequency synchronization among the user devices is not an easy task. Although uplink MU-MIMO was extensively discussed in IEEE 802.11ac, it was not included in this standard because of the concerns about the increase of network complexity and communication overhead induced by the time and frequency synchronization. IEEE 802.11ax is the first WLAN standard that includes uplink MU-MIMO. Different from 802.11ac, which is a CSMA-based distributed wireless network, 802.11ax is a centralized wireless network where the APs take the responsibility of user scheduling and OFDMA-based resource allocation. To enable uplink MU-MIMO, 802.11ax has specified its network-wide synchronization requirements: the timing offset should be less than $0.4 \mu\text{s}$ and the carrier frequency offset should be less than 350 Hz [3].

To achieve the specified time synchronization among user devices, 802.11ax borrowed the ideas from cellular networks (4G LTE) and proposed timing advance mechanism to adjust the transmit timing at each user device. Furthermore, to combat the residual $0.4 \mu\text{s}$ time offset, 802.11ax increases the cyclic prefix (CP) length of its OFDM symbols from $0.4 \mu\text{s}$ in 802.11ac to $0.8 \mu\text{s}$. The introduction of timing advance mechanism and the use of long CP will increase both network complexity and communication overhead. Moreover, to achieve the specified carrier frequency synchronization, user devices need to adjust their internal clock and frequency references by extracting offset information from the trigger frames that they get from the AP. This will result in an increase of computational complexity for user devices. Therefore, maintaining such strict time and frequency synchronization in WLANs is a luxury, which cannot be afforded by some wireless networks.

In this paper, we show that the time and frequency synchronization among user devices is not a necessity for uplink MU-MIMO in WLANs. We propose a practical uplink MU-MIMO solution which does not require time and frequency alignments of the signals from different user devices. The key component in our solution is a new PHY design (termed ForkPHY) for AP's receiver, which can decode the concurrent signals from multiple asynchronous user devices.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5770-8/18/06...\$15.00

<https://doi.org/10.1145/3209582.3209585>

Different from conventional MIMO PHY, ForkPHY decodes the signal from each user device separately. When decoding the signal from one user device, it simply treats the signals from other user devices as interference. Then, it employs a spatial filter to cancel the interference and recover the signal. Despite its simplicity, ForkPHY has surprisingly superior performance in decoding asynchronous signals from multiple user devices. Moreover, ForkPHY does not rely on any dedicated MAC protocols and thus can work with the existing user grouping protocols (e.g., [8, 32]).

We have built a prototype of our uplink MU-MIMO solution on USRP2-GNURadio testbed. To quantify the performance of ForkPHY, we use error vector magnitude (EVM) as the performance metric, which is defined in the IEEE 802.11 standards and has become an industry standard measurement for wireless devices. We measure the EVM performance of ForkPHY in two-user, three-user, and four-user uplink cases in ten different indoor scenarios. Experimental results show that, using ForkPHY, an M -antenna AP can concurrently serve M asynchronous stations ($2 \leq M \leq 4$). In addition, a real-time demo of asynchronous two-user uplink is presented in [28].

The remainder of the paper is organized as follows. In Section 2, we present the related work. In Section 3, we define the problem. In Section 4, we develop an asynchronous multi-user detection (AMUD) algorithm. Based on the AMUD algorithm, we design ForkPHY in Section 5 and discuss its applications in Section 6. Section 7 implements ForkPHY on wireless testbed and Section 8 evaluates the performance of ForkPHY using experimental results. Section 9 concludes the paper.

2 RELATED WORK

Uplink MU-MIMO. While there exist many results on uplink MU-MIMO, most of them focus on information-theoretic capacity exploration (see, e.g., [9, 13, 30]), user grouping and scheduling (see, e.g., [8, 32]), and uplink resource allocation (see, e.g., [14, 16, 17, 24, 25, 31]). However, all these research results were obtained under the assumption that the user devices are perfectly synchronized in both the time and frequency domains. Different from these results, our work revisits the time and frequency synchronization requirements in uplink MU-MIMO and aims at eliminating the synchronization requirements for distributed user devices.

Network-Wide Synchronization in LTE and WLANs. As the time and frequency synchronization among user devices is widely considered as a requisite for uplink MIMO, research and development efforts have been invested on achieving the synchronization requirements. In 4G LTE systems [19], eNBs take the advantage of GPS disciplined clocks to synchronize the carrier frequency of all user devices and maintain the time synchronization of arriving signals through timing advance mechanism. In IEEE 802.11ax [3], APs do not have GPS disciplined clocks and have to maintain system synchronization using dedicated MAC protocols. [12] points out that synchronization is one of the grand challenges in 802.11ax. [10] shows that the performance of uplink MU-MIMO is susceptible to the synchronization performance in 802.11ax. The high cost and susceptibility of network-wide synchronization motivate us to develop a new uplink MU-MIMO solution that can be used in asynchronous wireless networks.

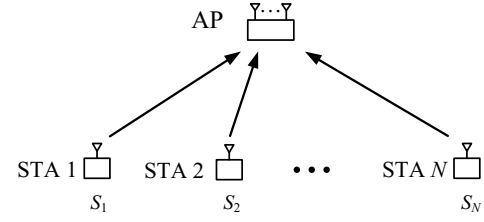


Figure 1: Uplink MU-MIMO in a WLAN.

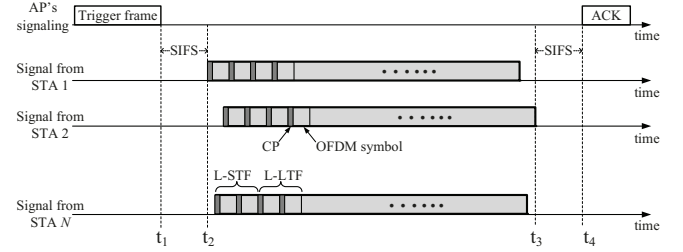


Figure 2: Uplink MU-MIMO MAC protocol and illustration of time misalignment of the received signals at the AP.

Asynchronous Multi-User Detection. Another research stream closely related to this work is asynchronous multi-user detection. While there is a large amount of work on multi-user detection (see, e.g., [1, 4, 6, 7, 15, 20, 21, 23, 26, 29]), most of them assume that the received signals are perfectly synchronized. Results of asynchronous multi-user detection remain rare. [21] is a pioneering work in this area. To decode the concurrent signals from asynchronous users, it developed a cross-layer solution by combining the PHY-layer ZF-SIC technique and a dedicated MAC-layer frame scheduling. Specifically, it uses a chain-decoding technique to estimate channel parameters and iteratively decode the signal frames with misaligned symbol timing and frequency offsets. However, this approach requires channel estimation, interference cancellation, signal reconstruction, signal subtraction, and signal detection. Moreover, it requires a dedicated MAC protocol to support sophisticated signal processing. These requirements make it impractical. In contrast, ForkPHY is a practical and MAC-independent PHY solution to enable asynchronous multi-user detection in WLANs.

3 PROBLEM STATEMENT AND MODELING

3.1 Problem Statement

Consider the uplink MU-MIMO in a WLAN as shown in Figure 1, which consists of one multi-antenna AP and a set of single-antenna user devices (a.k.a. stations or STAs). Denote M as the number of antennas on the AP. Denote N as the set of stations, with N being its cardinality (i.e., $N = |N|$). In our study, we assume that the number of antennas on the AP is greater than or equal to the number of stations, i.e., $M \geq N$. We note that in real-world WLANs, the set of stations participating in uplink MU-MIMO transmission is selected by upper-layer user scheduling algorithm. Therefore, this assumption can be easily fulfilled.

Uplink MU-MIMO Protocol. Figure 2 shows the uplink MU-MIMO protocol in our study. It consists of the following three steps. First, the AP sends a trigger frame to announce uplink MU-MIMO transmission. This trigger frame includes the identities of the set of stations which have been selected for uplink MU-MIMO transmission in this round. Second, upon receiving the trigger frame from the AP, the selected stations prepare their packets and send their packets to the AP. Third, the AP decodes the packets from the selected stations and sends an ACK/NACK packet to the stations.

Legacy Frame Format. We assume that the uplink MU-MIMO transmission uses IEEE 802.11 legacy frame format. As shown in Figure 2, a legacy frame is comprised of preamble, signal field, and data field. Particularly, the preamble consists of Legacy Short Training Field (L-STF) and Legacy Long Training Field (L-LTF), each of which has two OFDM symbols. We note that, in our uplink MU-MIMO solution, only the legacy preamble is used and the High-Throughput (HT) preamble is not required.

No Synchronization. We assume that there is no dedicated protocol used to synchronize the stations in uplink MU-MIMO. In the time domain, since the signals from different stations have different propagation delays and the stations' transmit timing is not disciplined, the received signals at the AP will be misaligned in the time, as illustrated in Figure 2. Typically, the misalignment gap is longer than the time duration of CP. As a result, the signals from different stations constitute interference to each other, and conventional multi-user detection approaches such as MMSE and ZF are not able to decode the signals. Moreover, in the frequency domain, since each station is driven by its own oscillator, the signals from different stations have different carrier frequency offsets with respect to the AP, making it even more challenging to decode the signals at the AP.

Our Objective. Our objective is to develop a practical PHY solution for the AP's receiver so that it can successfully decode the asynchronous signals from the stations.

3.2 Mathematical Modeling

Consider uplink MU-MIMO in Figure 1. At STA i , denote S_i as its transmit signal. At the AP, denote $\mathbf{Y} = [Y_1, Y_2, \dots, Y_M]^T$ as its received signal vector, where Y_j is the signal from its j th antenna. Denote $\mathbf{W} = [W_1, W_2, \dots, W_M]^T$ as the noise vector at the receiver, where W_j is the noise from its j th antenna. Denote $\mathbf{H}_i$ as the channel vector between STA i and the AP.

If the stations were perfectly synchronized, then the uplink MU-MIMO would be equivalent to a $N \times M$ point-to-point MIMO and its mathematical modeling could be written as:

$$\mathbf{Y} = \sum_{i \in \mathcal{N}} \mathbf{H}_i S_i + \mathbf{W}. \quad (1)$$

However, WLAN is a distributed network and we assume that there is no protocol used to synchronize the stations. Due to the lack of time and frequency synchronization among the stations, (1) is an incorrect mathematical model to describe uplink MU-MIMO. As a result, conventional MIMO detection algorithms such as MMSE and ZF are not able to decode the asynchronous signals at the AP.

We propose a new detection algorithm to decode the asynchronous signals at the AP. Unlike MMSE and ZF MIMO detection

algorithms, which decode the signals jointly, our detection algorithm decodes the asynchronous signals *separately*. When decoding the signal from one station, we simply treat the signals from other stations as interference. Then, we employ a spatial filter to cancel the interference and recover the desired signal. In what follows, we focus on decoding the signal from station $i \in \mathcal{N}$. Decoding the signals from other stations shall follow the same token.

In order to decode the signal from station i , we model the received signals at the AP as follows:

$$\mathbf{Y} = \mathbf{H}_i S_i + \sum_{k \in \mathcal{N}, k \neq i} \mathbf{H}_k I_k + \mathbf{W}, \quad (2)$$

where S_i is the signal from station i and I_k is the interfering signal from station k ($k \neq i$). It is worth pointing out that, since the time and frequency offsets differ from station to station and vary from time to time, it is extremely hard to estimate I_k in (2) based on the original signal S_k . Therefore, the AP has to decode the signal S_i without knowledge of interference I_k , $k \in \mathcal{N} \setminus \{i\}$. This is the main challenge in asynchronous uplink multi-user detection.

4 AN ASYNCHRONOUS MULTI-USER DETECTION (AMUD) ALGORITHM

In this section, we develop an AMUD algorithm for AP to decode the asynchronous signals from the stations. The core of the algorithm is a linear spatial filter (termed AMUD filter) for each station, which aims to recover the signal from station i and cancel the unknown interference from other stations. In what follows, we first present AMUD filter design and then analyze its performance.

4.1 AMUD Filter Design

At the AP, we employ a linear spatial filter to decode the signal S_i from station i . Denote $\mathbf{P}_i$ as the linear spatial filter (an $M \times 1$ complex vector). Denote $\hat{S}_i$ as the estimated signal from station i . Then we have:

$$\hat{S}_i = \mathbf{P}_i^H \mathbf{Y}, \quad (3)$$

where $(\cdot)^H$ represents conjugate transpose.

Based on (3), the mean squared error (MSE) can be written as:

$$\begin{aligned} \text{MSE} &= \mathbb{E}[|\hat{S}_i - S_i|^2] = \mathbb{E}[\|\mathbf{P}_i^H \mathbf{Y} - S_i\|^2] \\ &= \mathbf{P}_i^H \mathbb{E}[\mathbf{Y} \mathbf{Y}^H] \mathbf{P}_i + \mathbb{E}[S_i S_i^H] - \mathbb{E}[\mathbf{P}_i^H \mathbf{Y} S_i^H] - \mathbb{E}[S_i \mathbf{Y}^H \mathbf{P}_i], \end{aligned} \quad (4)$$

where $\mathbb{E}(\cdot)$ represents statistical expectation. This equation is actually a quadratic function of $\mathbf{P}_i$. To minimize MSE, we can take the gradient with respect to $\mathbf{P}_i$. The optimal filter $\mathbf{P}_i$ can be obtained by setting the gradient to zero, which we show as follows:

$$\frac{\partial \text{MSE}}{\partial \mathbf{P}_i} = 2\mathbb{E}[\mathbf{Y} \mathbf{Y}^H] \mathbf{P}_i - 2\mathbb{E}[\mathbf{Y} S_i^H]. \quad (5)$$

By setting $\frac{\partial \text{MSE}}{\partial \mathbf{P}_i}$ to zero, we can obtain the optimal filter by

$$\mathbf{P}_i = \mathbb{E}[\mathbf{Y} \mathbf{Y}^H]^{-1} \mathbb{E}[\mathbf{Y} S_i^H]. \quad (6)$$

Equation (6) is the optimal design of $\mathbf{P}_i$ in the sense of minimizing MSE. Now the question is how to calculate $\mathbb{E}[\mathbf{Y} \mathbf{Y}^H]$ and $\mathbb{E}[\mathbf{Y} S_i^H]$ in (6). For this question, we take advantage of the reference symbols in wireless systems (e.g., the preamble OFDM symbols in legacy 802.11 frame, i.e., L-STF and L-LTF). Denote L as the number of reference

symbols in a frame. Denote $[\tilde{S}_i(1), \tilde{S}_i(2), \dots, \tilde{S}_i(L)]$ as the reference symbols at station i . Denote $[\tilde{Y}(1), \tilde{Y}(2), \dots, \tilde{Y}(L)]$ as the received reference symbols at the AP, which also include interfering signals from other stations. Then we approach the statistic expectation using the averaging operation over the set of reference symbols. Specifically, we estimate $\mathbb{E}[\mathbf{Y}\mathbf{Y}^H]$ and $\mathbb{E}[\mathbf{Y}\mathbf{S}_i^H]$ in (6) as follows:

$$\mathbb{E}[\mathbf{Y}\mathbf{Y}^H] \leftarrow \frac{1}{L} \sum_{l=1}^L \tilde{Y}(l)\tilde{Y}(l)^H, \quad (7)$$

$$\mathbb{E}[\mathbf{Y}\mathbf{S}_i^H] \leftarrow \frac{1}{L} \sum_{l=1}^L \tilde{Y}(l)\tilde{S}_i(l)^H, \quad (8)$$

where $\leftarrow$ operator represents value estimation.

Based on (7) and (8), we construct the AMUD filter $\mathbf{P}_i$ as follows:

$$\mathbf{P}_i = \left[\sum_{l=1}^L \tilde{Y}(l)\tilde{Y}(l)^H \right]^{-1} \left[\sum_{l=1}^L \tilde{Y}(l)\tilde{S}_i(l)^H \right]. \quad (9)$$

We now summarize our AMUD algorithm as follows: *To decode the signal from station i at the AP in Figure 1, we first construct AMUD filter $\mathbf{P}_i$ using (9) and then estimate the signal $\hat{S}_i$ using (3). The same algorithm is used to decode the signals from other stations.*

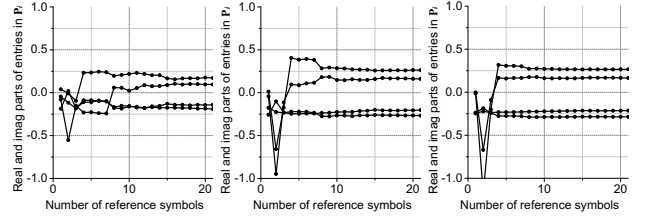
We have the following observations on the AMUD algorithm. First, the AMUD algorithm requires neither channel information $\mathbf{H}_i$ nor the knowledge of interference I_k . It only needs to know the reference symbols (preamble) at station i , i.e., $[\tilde{S}_i(1), \tilde{S}_i(2), \dots, \tilde{S}_i(L)]$. Second, $\tilde{Y}(l)$ in (9) includes both the reference signals from station i and the interfering signals from other stations. This means that a clean (interference-free) piece of reference signals is not required to decode the signal. In other words, the reference signals at the stations can be sent concurrently. Due to these special properties, the AMUD algorithm is particularly suited for multi-user detection in asynchronous wireless networks.

4.2 Performance Analysis

SINR Performance. We analyze the SINR performance of our proposed AMUD algorithm by comparing it with the conventional MMSE MIMO detection algorithm, which has been well studied and regarded as one of the most superior detection algorithms in MIMO communications [22, Ch. 7]. Consider the same network in Figure 1. Let us assume that all the stations in the network are perfectly synchronized. Then, the uplink MU-MIMO is equivalent to point-to-point MIMO and the AP can use conventional MMSE detector to decode the signals from the stations. The MMSE detector can be written as $\mathbf{P}_{\text{mmse}} = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H + \mathbf{R}_w)^{-1}$, where $\mathbf{H}$ is the compound channel and $\mathbf{R}_w$ is noise correlation matrix [22, Ch. 7]. Contrasting the AMUD algorithm to the MMSE detection algorithm, we have the following lemma:

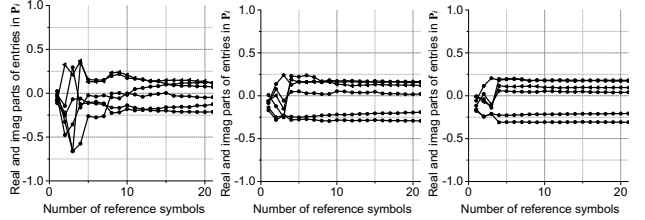
LEMMA 4.1. *If reference symbols are sufficient, then the AMUD algorithm in an asynchronous network has the same performance as the conventional MMSE MIMO detection algorithm in a synchronous network.*

Proof: Consider the AP in Figure 1. The AMUD algorithm decodes the signals from each individual station separately. Based on (6), the AMUD filter for decoding the signals from station i can be written as $\mathbf{P}_i = \mathbb{E}[\mathbf{Y}\mathbf{Y}^H]^{-1}\mathbb{E}[\mathbf{Y}\mathbf{S}_i^H]$. By putting the N AMUD filters together,



(a) SNR=0dB case. (b) SNR=10dB case. (c) SNR=20dB case.

Figure 3: Convergence of AMUD filter $\mathbf{P}_i$ in (9) over the number of reference symbols in a WLAN that consists of one two-antenna AP and two single-antenna stations.



(a) SNR=0dB case. (b) SNR=10dB case. (c) SNR=20dB case.

Figure 4: Convergence of AMUD filter $\mathbf{P}_i$ in (9) over the number of reference symbols in a WLAN that consists of one three-antenna AP and three single-antenna stations.

the AMUD matrix, denoted as $\mathbf{P}_{\text{amud}}$, can be written as $\mathbf{P}_{\text{amud}} = [\mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_N] = \mathbb{E}[\mathbf{Y}\mathbf{Y}^H]^{-1}\mathbb{E}[\mathbf{Y}\mathbf{S}^H]$, where $\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N]$. Then, we have

$$\begin{aligned} \mathbf{P}_{\text{amud}} &= \mathbb{E}[\mathbf{Y}\mathbf{Y}^H]^{-1}\mathbb{E}[\mathbf{Y}\mathbf{S}^H] = [\mathbf{H}\mathbf{H}^H + \mathbf{R}_w]^{-1}\mathbf{H} \\ &= (\mathbf{H}^H[\mathbf{H}\mathbf{H}^H + \mathbf{R}_w]^{-1})^H = (\mathbf{P}_{\text{mmse}})^H. \end{aligned} \quad (10)$$

Recall that at the AP, the AMUD algorithm decodes the signals by $\hat{\mathbf{S}} = (\mathbf{P}_{\text{amud}})^H \mathbf{Y}$ and the conventional MMSE algorithm decodes the signals by $\hat{\mathbf{S}} = (\mathbf{P}_{\text{mmse}}) \mathbf{Y}$. Therefore, we conclude that the two algorithms have the same performance. $\square$

Based on Lemma 4.1 and the properties of MMSE algorithm, we have the following corollary:

COROLLARY 4.2. *If reference symbols are sufficient and noise is negligible, then the AMUD algorithm can perfectly recover the signals from the asynchronous stations (i.e., $\hat{S}_i = S_i$ for $i \in N$).*

Convergence over Reference Symbols. Lemma 4.1 and Corollary 4.2 show the superior performance of the AMUD algorithm in asynchronous networks when the reference symbols are sufficient. In real-world WLANs, the amount of reference symbols (preamble symbols) in a frame is limited. Therefore, it is important to study the convergence speed of the AMUD filter over the number of reference symbols. As an instance, Figure 3 presents the convergence speed of the AMUD filter in (9) over the number of reference symbols in a WLAN with two stations. From the simulation results we can see that the AMUD filter converges in 15 reference symbols when SNR = 0 dB, in 10 reference symbols when SNR = 10 dB, and in 5 reference symbols when SNR = 20 dB. Figure 4 presents the convergence results when the WLAN has three stations, and we

have the similar observations. Based on the simulation results, we conclude that the AMUD algorithm converges at a very fast speed.

In addition to simulation results, we will provide experimental results in Section 8 to show that the preamble (i.e., L-STF and L-LTF) in a legacy 802.11 frame are sufficient for the AMUD algorithm to decode the asynchronous signals.

Computational Complexity. From (9) we can see that the AMUD algorithm involves matrix multiplication and one-time matrix inverse manipulation. All these manipulations are linear operations. The dimension of the matrix is the number of antennas at the AP, which is typically small (less than or equal to eight per 802.11ac standard). Hence, the AMUD algorithm has similar computational complexity as ZF MIMO detector, which has been widely implemented in real-world wireless systems.

4.3 AMUD in OFDM-MIMO Network

AMUD in OFDM Networks. In a broadband network, OFDM modulation has been used to divide the broadband channel into many narrow-band parallel channels. Each OFDM subcarrier corresponds to one narrow-band channel. To decode the signals in a broadband network, we employ the AMUD algorithm on each of its OFDM subcarriers. Specifically, for each individual subcarrier, we use (9) to calculate its AMUD filter and then use (3) to decode the desired signal.

Multiple Antennas at Station. The AMUD algorithm was developed based on the uplink model in Figure 1, where each station has one antenna. But in real-world WLANs, the stations may have multiple antennas. Does the AMUD algorithm still work? The answer is yes and we explain it as follows. For a station having multiple antennas, it can use its multiple antennas for either spatial diversity or spatial multiplexing. Both cases are supported by IEEE 802.11 standards. If a station uses its multiple antennas for spatial diversity, then its multiple antennas can be regarded as one combined antenna from AP's point of view [22, Ch. 7]. Therefore, the AMUD algorithm can completely handle this case. If a station uses its multiple antennas for spatial multiplexing, then the station can be treated as multiple virtual single-antenna stations from the AP's point of view. Therefore, the AMUD algorithm can also handle this case. To sum up, as long as the number of antennas on the AP is no less than that on the stations, the AMUD algorithm can always be used to decode the asynchronous signals from the stations, regardless of the number of antennas on each individual station.

5 FORKPHY DESIGN

Based on the AMUD algorithm, we design ForkPHY — a new PHY for AP's receiver. Figure 5 shows the block diagram of ForkPHY. As shown in the figure, ForkPHY has N parallel baseband signal processing paths, each of which intends to decode the signal from one station. Each path consists of the following modules: a sync module, a FFT module, and an AMUD module, and an array of registers for AMUD filters. Compared to conventional MIMO PHY, ForkPHY has the following new components: the sync module, the AMUD module, and the registers of AMUD filters. In what follows, we present the new components in the i th path ($i \in N$).

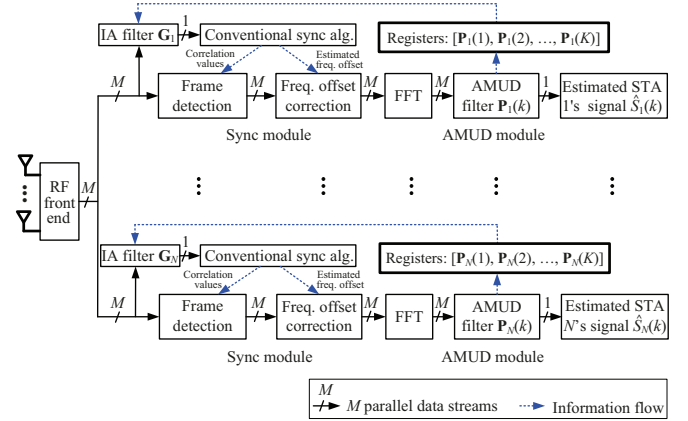


Figure 5: A new PHY for AP's receiver (ForkPHY).

5.1 Registers of AMUD Filters

As shown in Figure 5, each path i has an array of registers. These registers are used to store the values of AMUD filters $[P_i(1), P_i(2), \dots, P_i(K)]$, where K is the number of valid subcarriers in OFDM modulation and $P_i(k)$ is the MUD filter for decoding signal from station i on subcarrier k . The registers have a timer. Whenever the AMUD module has calculated the values of $[P_i(1), P_i(2), \dots, P_i(K)]$, it updates the registers immediately. At the same time, the register timer is reset and runs again.

For the AMUD filters in the registers, if the timer has not reached a predefined threshold, then they are considered valid; otherwise, they are considered invalid. The time threshold should be set based on the channel coherent time, which is determined by moving speed of the station as well as the changing speed of surrounding environment. Empirically, we find setting the threshold to 10 ms is a good option for both stationary and mobile scenarios in indoor environments.

5.2 Sync Module

The sync module in ForkPHY is a signal processing block for signal synchronization on the AP. It is completely different from the synchronization that we discussed in Section 1, which refers to the network-wide time and frequency synchronization among all the stations.

The sync module in ForkPHY has two functionalities: time synchronization and frequency synchronization. Time synchronization is to detect bursty frame by exploiting auto or cross correlation of the signal stream in the time domain. Frequency synchronization is to estimate and correct the carrier frequency offset between AP and station. In our design, each path has an independent sync module. This is because the signals from different stations have different time and frequency offsets when arriving at the AP. Synchronization in each path is challenging, as it must be done in the presence of interfering signals from other stations. To address this challenge, we first use a spatial filter to alleviate interference in the time domain and then employ conventional sync algorithm to synchronize the signal from a particular station.

As shown in Figure 5, the sync module consists of four blocks. (i) *An interference-alleviation (IA) filter G_i* : This filter is an M -dimensional complex vector. Its input is M signal streams and its output is one signal stream. It aims to alleviate the interference from other stations in the time domain so as to improve the synchronization performance. (ii) *Conventional sync algorithm*: This block employs a conventional correlation-based synchronization algorithm (e.g., [2, 27]) to estimate the time and frequency offsets based on the interference-alleviated signal stream. (iii) *Frame detection*: Based on the estimated time offset, this block identifies the frame from the M input signal streams. (iv) *Frequency offset correction*: Based on the estimated value of frequency offset, this block corrects the frequency offset for the M input signal streams.

In this sync module, IA filter G_i is the key component and the other three blocks have been well studied. In what follows, we present our design of the IA filter by two cases.

Case I: AMUD Filters in Registers are Valid. If the AMUD filter values in the registers are valid, then we construct the IA filter by

$$G_i = \frac{1}{2A+1} \sum_{k=K/2-A}^{K/2+A} P_i(k), \quad (11)$$

where A is the number of neighboring subcarriers to be averaged and its value should be empirically set (e.g., 0, 1 or 2). It should be noted that $K/2$ in (11) is the central subcarrier index in the OFDM modulation. Regarding the performance of this filter, we have the following lemma:

LEMMA 5.1. *If the channels between stations and AP are frequency-flat and the noise is negligible, then the IA filter in (11) can completely cancel interfering signals from other stations.*

The argument of this lemma is straightforward based on Corollary 4.2 and therefore we omit it. Lemma 5.1 shows the efficacy of the IA filter in an ideal scenario. In practice, although the channels are not frequency-flat, the frequency responses of neighboring OFDM subcarriers are typically highly correlated. Therefore, the IA filter in (11) can significantly alleviate the interference from other stations and, as a result, the performance of the sync module can be significantly improved.

Case II: AMUD Filters in Registers are Not Valid. Since the AMUD filters in the registers are invalid, we cannot use the AMUD filters to construct the IA filter. In this case, we borrow the idea from the Multiple Signal Classification (MUSIC) algorithm by projecting the signals into orthogonal subspaces. Specifically, we use a left-singular vector of the signals as the IA filter. Denote $y(n)$ is the M data streams from the M antennas. Denote N_s as the number of signal samples for synchronization. Denote $\mathbf{U}$ as the left unitary matrix and $\mathbf{V}$ as the right unitary matrix. Then we have:

$$[\mathbf{U} \Sigma \mathbf{V}] = \text{svd} \left(\sum_{n=1}^{N_s} y(n)y(n)^H \right). \quad (12)$$

Based on the resulting left unitary matrix $\mathbf{U}$, we construct the IA filter as follows:

$$G_i = \mathbf{U}(m), \quad (13)$$

where $\mathbf{U}(m)$ represents the m th column of matrix $\mathbf{U}$. Given that $\mathbf{U}$ has M columns, the next question is which of the M columns we should use in (13). For this question, we resort to the exhaustive

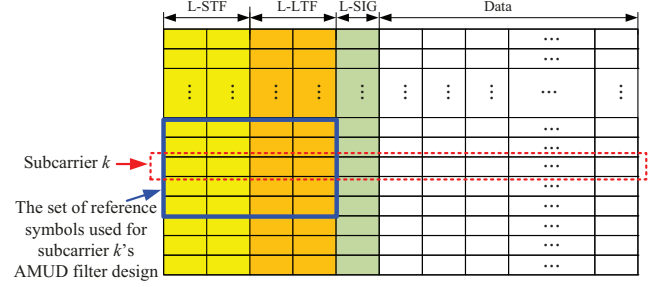


Figure 6: Illustration of $\mathcal{P}(k)$ in legacy 802.11 frame.

search approach. That is, we try each column of $\mathbf{U}$ and choose the one that results to the best cross correlation performance. It should be noted that the exhaustive search only needs to be done once for a long period of time and therefore the computational complexity would not be a problem in practice. For this IA filter, we have the following lemma:

LEMMA 5.2. *If the channels between stations and AP are frequency-flat and the noise is negligible, then the IA filter in (13) can completely cancel interfering signals from other stations.*

The argument and interpretation of Lemma 5.2 are similar to those of Lemma 5.1. We therefore omit it.

5.3 AMUD Module

As shown in Figure 5, in each data path, once a signal frame has been detected and the frequency offset has been corrected, the signal streams are fed into the FFT module, which converts each signal stream from the time domain to the frequency domain. For the resulting frequency-domain signals, we employ the AMUD algorithm on each subcarrier to decode the original signals from the station. Specifically, we first use (9) to compute the AMUD filter and then use (3) to estimate the original signal on each subcarrier.

As we explained in Section 4, the design of the AMUD filter needs reference signals. The more; the better. Now the question is: for each subcarrier, which reference symbols can be used for the design of its AMUD filter? For this question, we illustrate our solution in Figure 6. That is, for the design of subcarrier k 's AMUD filter, we use the reference symbols not only on that subcarrier but also on its neighboring subcarriers. The rationale behind this design is that the channels on the neighboring subcarriers are highly correlated in practice.

Denote $\mathcal{P}_k$ as the set of reference symbols that are used for subcarrier k 's AMUD filter design. As illustrated in Figure 6, we have $\mathcal{P}_k = \{(l, k') : 1 \leq l \leq 4; k-2 \leq k' \leq k+2\}$, where $1 \leq l \leq 4$ means the OFDM symbols in L-STF and L-LTF fields, and $k-2 \leq k' \leq k+2$ means the neighboring two subcarriers. Note that we use 2 neighboring subcarriers as an example. The number of neighboring subcarriers used for $\mathcal{P}_k$ should be determined upon the channel correlation, which can be empirically set. Denote $P_i(k)$ as the AMUD filter on subcarrier k in ForkPHY's i th path. Based on (9), the AMUD filter for decoding signals on subcarrier k can be

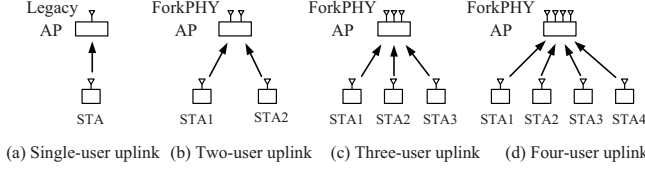


Figure 7: Single-user uplink and multi-user uplink. Single-user uplink in (a) serves as the performance benchmark.

written as:

$$\mathbf{P}_i(k) = \left[\sum_{(l,k') \in \mathcal{P}_k} \mathbf{Y}(l,k') \mathbf{Y}(l,k')^H \right]^{-1} \left[\sum_{(l,k') \in \mathcal{P}_k} \mathbf{Y}(l,k') S_i(l,k')^H \right], \quad (14)$$

where $\mathbf{Y}(l,k)$ is the vector of signals that the AP receives on subcarrier k of the l th OFDM symbol in a frame, and $S_i(l,k)$ is the signal that station i sends on subcarrier k of the l th OFDM symbol in a frame.

Denote $\hat{S}_i(l,k)$ as the estimated version of the signal from station i on subcarrier k of the l th OFDM symbol. Then, based on (3) and (14), we estimate the original signal by

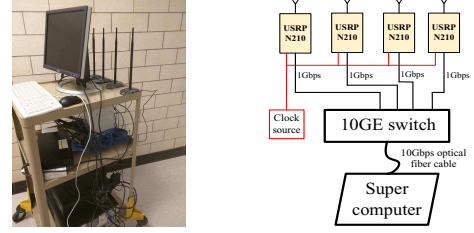
$$\hat{S}_i(l,k) = \mathbf{P}_i(k)^H \mathbf{Y}(l,k). \quad (15)$$

6 ASYNCHRONOUS UPLINK MU-MIMO SOLUTION

The proposed asynchronous uplink MU-MIMO solution is comprised of two components: ForkPHY for AP's receiver and the MAC protocol in Figure 2. Since the MAC protocol is straightforward, ForkPHY is the core of the asynchronous uplink MU-MIMO solution. In what follows, we offer discussions on ForkPHY.

Backward Compatibility: ForkPHY is a backward compatible PHY design for AP's receiver. Its backward compatibility is reflected in the two aspects. First, it can support legacy stations (802.11n/ac devices) in single-user uplink. This means that ForkPHY AP can maintain its services for millions of legacy stations in WLANs. Second, the asynchronous uplink MU-MIMO solution does not require any PHY modification on the stations. If a legacy station wants to participate in uplink MU-MIMO, it only needs to modify its MAC protocol stack through upgrading its firmware or driver, which is doable in practice.

Implementation Complexity: ForkPHY is a lightweight PHY for AP's receiver which is amenable to practical implementation. Compared to conventional MIMO PHY [22], ForkPHY has two new modules: sync and AMUD modules (see Figure 5). The sync module in ForkPHY reuses most part of the sync algorithm in conventional MIMO receiver. In the worst case, it needs an extra SVD operation for each frame compared with conventional sync algorithm. The AMUD module involves a matrix inverse operation in (14). It has similar computational complexity as the MMSE and ZF MIMO detectors. With the continuous advances in semiconductor technology, we expect no computational issues in implementing ForkPHY on APs.



(a) ForkPHY AP. (b) AP's schematic diagram.

Figure 8: Our prototype of ForkPHY AP.

7 IMPLEMENTATION

Figure 7 shows the network topologies that we consider for the performance evaluation of our uplink MU-MIMO solution.

Prototype of AP: We have built an AP using four USRP N210 devices [18], one OctoClock-G CDA-2990 [18], one 10GbE-Switch, and the GNURadio software package [5]. Figure 8 shows ForkPHY AP and its schematic diagram. We implement ForkPHY in a super-computer for the AP's baseband signal processing. This AP can support up to four antennas. It can also be used as a three-antenna AP or a two-antenna AP by simply turning off one or two of its antennas.

Prototype of Stations: We have built four independent stations using four USRP N210 devices. These four stations are driven by their own oscillators and no synchronization protocol is used. Each of them has a single antenna. The carrier frequency is set to 2.48 GHz and the bandwidth is set to 10 MHz. Measurement results show that the four stations have different frequency offsets with respect to the prototyped AP: 3.2 kHz, 0.1 kHz, -2.3 kHz, and 1.3 kHz. The stations run a simplified PHY layer of 802.11n in legacy mode (see the frame structure in Figure 6). Each OFDM symbol has 64 subcarriers, with 48 of them being used for payloads and 4 being used for pilots. QPSK modulation is used for data transmission.

Prototype of Protocol: We implemented the uplink MU-MIMO protocol in Figure 2, except the ACK part. In our testbed, the trigger frame was implemented using a NDPA and the uplink frame has 20 OFDM symbols in total (see frame structure in Figure 6). To mimic large propagation delays in large-size WLANs, we added additional transmitting delays for STA 2, STA 3, and STA 4, which are random variables drawn uniform distribution in $[0\mu s, 2\mu s]$, $[2\mu s, 4\mu s]$, and $[4\mu s, 6\mu s]$, respectively. These additional transmitting delays also ensure that the time offsets of the received signal frames at the AP are greater than the CP duration of OFDM symbol.

8 PERFORMANCE EVALUATION

We evaluate the performance of ForkPHY in three cases: (i) two-antenna AP in two-user uplink as shown in Figure 7(b); (ii) three-antenna AP in three-user uplink as shown in Figure 7(c); and (iii) four-antenna AP in four-user uplink as shown in Figure 7(d). In our evaluation, we use the legacy single-antenna AP in single-user uplink in Figure 7(a) as the performance benchmark.

8.1 Experimental Setup

We measure the performance of ForkPHY in an office building as shown in Figure 9. The AP is placed at the location marked "AP" in

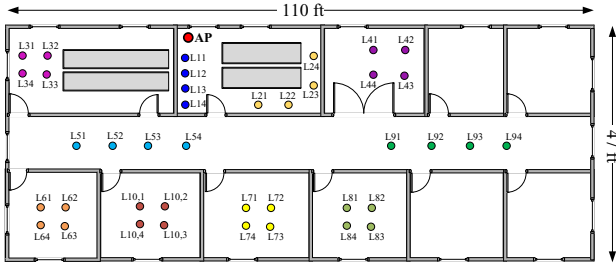
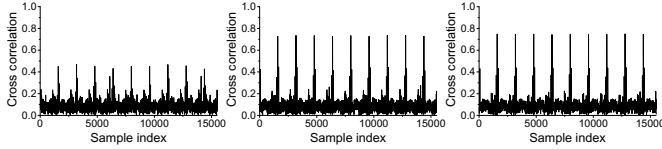
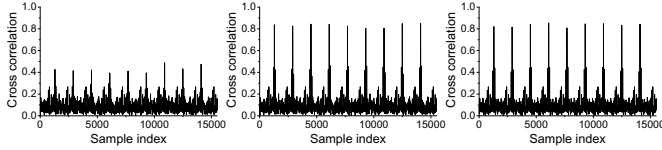


Figure 9: Ten test scenarios in an office building (L_{ki} is the location of STA i in test scenario k).



(a) No IA filter (b) Use IA filter in (11) (c) Use IA filter in (13)

Figure 10: Results of cross correlation at ForkPHY AP when detecting signals from STA1.



(a) No IA filter (b) Use IA filter in (11) (c) Use IA filter in (13)

Figure 11: Results of cross correlation at ForkPHY AP when detecting signals from STA2.

Figure 9. The four stations are placed at one of the 10 test scenarios in Figure 9. Specifically, in test scenario k , station i is placed at location L_{ki} , $1 \leq k \leq 10$ and $1 \leq i \leq 4$. In all the scenarios, the stations use the same transmit power, which is 17 dBm unless otherwise stated.

8.2 Performance Metric

We use error vector magnitude (EVM) as the performance metric, which is defined in the IEEE 802.11 standards and has become an industry standard measurement for Wi-Fi devices. Mathematically, $\text{EVM (dB)} = 10 \log_{10} \left(\frac{\mathbb{E}(|S_i - \hat{S}_i|^2)}{\mathbb{E}(|S_i|^2)} \right)$, where S_i and $\hat{S}_i$ are the original and estimated signals, respectively. For QPSK modulation, once EVM is obtained, its RawBER can be calculated using the closed-form expressions: $\text{RawBER} = 2Q(\gamma) - Q(\gamma)^2$, where $\gamma = 10^{-\text{EVM(dB)}/10}$ and $Q(\cdot)$ is the q -function.

Per IEEE 802.11n/ac standards [11], $\text{EVM} = -10$ dB is sufficiently small for QPSK modulation. Therefore, in our evaluation, we use $\text{EVM} = -10$ dB as the threshold to judge whether the signals from the asynchronous stations are successfully decoded at the AP.

8.3 A Case Study

We use the two-user uplink in Figure 7(b) as a case study to show the performance of ForkPHY. We place STA1 at L_{11} and STA2 at

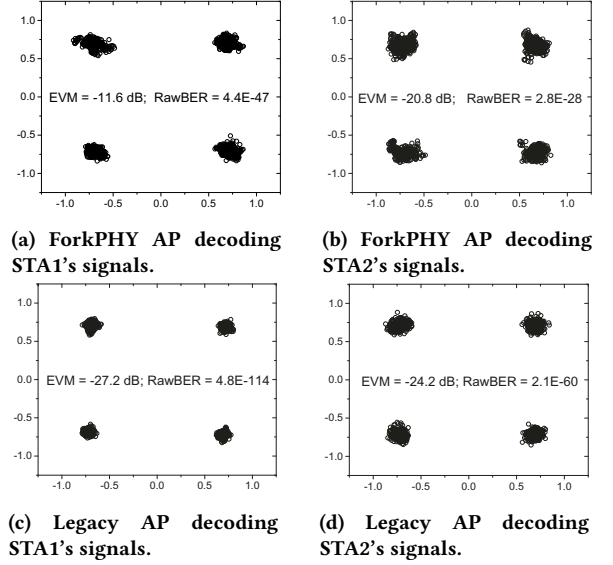
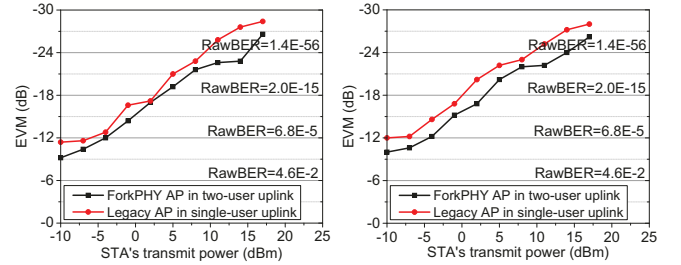


Figure 12: ForkPHY AP in two-user uplink versus legacy AP in single-user uplink.



(a) AP decoding STA1's signal. (b) AP decoding STA2's signal.

Figure 13: EVM Performance of AP when decoding the signal from its stations.

L_{12} in Figure 9. In this case, a real-time demo with video traffic is provided on our website [28] and the details of experimental results are presented in the following.

Performance of Sync Module. We first evaluate the performance of the sync module in ForkPHY. Recall that the core of the sync module is two IA filters, which are constructed in (11) and (13). We evaluate their impacts on ForkPHY's cross correlation performance. In our experiments, the cross-correlation results are obtained by correlating the received signal with a local L-LTF signal at the AP.

Figure 10 and Figure 11 present ForkPHY's cross correlation results when detecting signals from STA1 and STA2, respectively. Comparing Figure 10(a) and Figure 10(b), we can see that the IA filter in (11) significantly improves the time synchronization performance of ForkPHY. Comparing Figure 10(a) and Figure 10(c), we can see that the IA filter in (13) also significantly improves the time synchronization of ForkPHY. We have the same observations in Figure 11.

With these two IA filters, we also have the observation that the frequency offsets are accurately estimated and corrected at the AP.

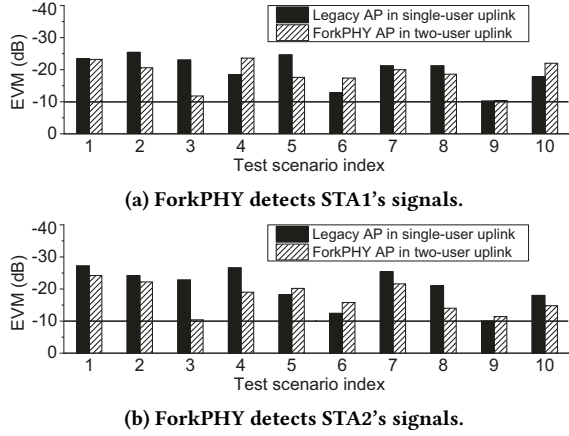


Figure 14: ForkPHY AP in two-user uplink versus legacy AP in single-user uplink.

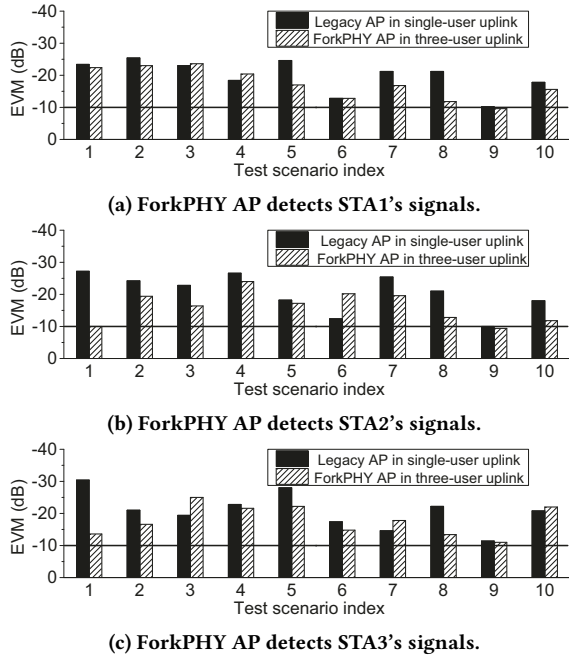


Figure 15: ForkPHY AP in three-user uplink versus legacy AP in single-user uplink.

We do not explicitly present the frequency synchronization results because the performance of frequency synchronization is directly reflected by the EVM performance of constellation (shown below). **Performance of AMUD Algorithm.** We now evaluate the performance of the AMUD algorithm at ForkPHY AP. Figure 12 presents the constellation diagram comparison between ForkPHY AP in two-user uplink and legacy AP in single-user uplink. The calculated EVM and RawBER are shown in the figure. From Figure 12(a–b), it is evident to see that ForkPHY AP can successfully decode the signals from the two asynchronous stations. Figure 12(c–d) serve as performance benchmarks for the two stations. Comparing Figure 12(a–b)

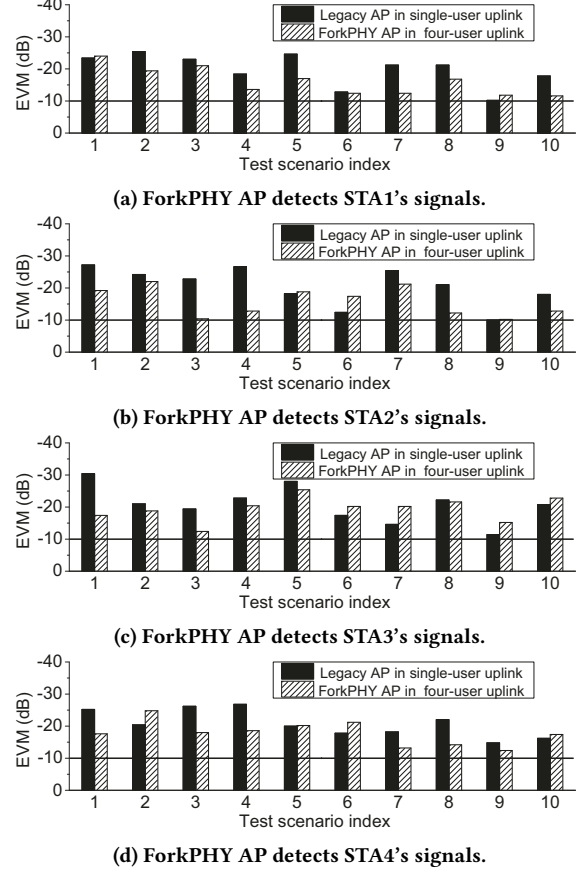


Figure 16: ForkPHY AP in four-user uplink versus legacy AP in single-user uplink.

with Figure 12(c–d), we can see that the EVM degradation is less than 4 dB for both stations.

We now evaluate the impact of station's transmit power on the performance of ForkPHY AP. Again, we use legacy AP in single-user uplink as the performance benchmark. Figure 13 shows our experimental results. From the results we can see that, compared to legacy AP in single-user uplink, ForkPHY AP can provide similar performance to the stations (with about 3 dB EVM degradation).

8.4 Complete Experimental Results

We study the performance of ForkPHY AP in the two-user uplink, three-user uplink, and four-user uplink cases. In each case, we measure the EVM of constellation in the 10 different test scenarios as shown in Figure 9. In the evaluation, we use the legacy AP in single-user uplink (see Figure 7(a)) as the performance benchmark. **Two-User Uplink Case.** As shown in Figure 7(b), the ForkPHY AP has two antennas and it serves as two asynchronous stations, which are placed at L_{k1} and L_{k2} in test scenario k ($1 \leq k \leq 10$). Figure 14 presents the EVM performance of ForkPHY AP when detecting the signals from the two asynchronous stations. From Figure 14(a) we can see that ForkPHY AP can successfully decode the signals from STA1 as the EVM in all test scenarios is less than -10 dB. We can also see that the ForkPHY AP offers similar performance to STA1 as

the legacy AP does in single-user uplink. Figure 14(b) presents the EVM performance of the ForkPHY AP when detecting the signals from STA2. From the figure we have the same observations for STA2.

Three-User Uplink Case. As shown in Figure 7(c), the ForkPHY AP has three antennas and it serves three asynchronous stations, which are placed at L_{k1} , L_{k2} , and L_{k3} in test scenario k ($1 \leq k \leq 10$). Figure 15 presents the EVM performance of ForkPHY AP when detecting the signals from the three asynchronous stations. From Figure 15 we can see that ForkPHY AP can successfully decode the signals from the three asynchronous stations (since their EVM is less than -10 dB).

Four-User Uplink Case. As shown in Figure 7(d), the ForkPHY AP has four antennas and it serves four asynchronous stations, which are placed at L_{k1} , L_{k2} , L_{k3} , and L_{k4} in test scenario k ($1 \leq k \leq 10$). Figure 16 presents the EVM performance of ForkPHY AP when detecting the signals from the four asynchronous stations. From Figure 16 we can see that ForkPHY AP can successfully decode the signals from the four asynchronous stations.

Summary of Observations. We summarize our observations on the experimental results. For ForkPHY AP with M antennas ($2 \leq M \leq 4$), it can successfully decode the concurrent signals from M asynchronous stations. Furthermore, its performance of decoding the signals from each station is comparable with the legacy single-antenna AP in single-user uplink.

9 CONCLUSION

In this paper, we show that the network-wide time and frequency synchronization among user devices is not a necessary for uplink MU-MIMO in WLANs. We proposed a practical solution which can enable asynchronous uplink MU-MIMO in WLANs. The enabler is ForkPHY, which is a new PHY design for AP's receiver that can decode the concurrent signals from multiple asynchronous user devices. Different from conventional MIMO PHY, ForkPHY does not require channel estimation when decoding the asynchronous signals. We built a prototype of ForkPHY on USRP2-GNURadio testbed. Experimental results show that ForkPHY AP with M antennas can successfully decode the concurrent signals from M asynchronous user devices ($2 \leq M \leq 4$).

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their valuable comments and feedback. This project was partially supported by the NSF under Grant CNS-1717840.

REFERENCES

- [1] Daniele Angelosante, Ezio Biglieri, and Marco Lops. 2009. Multiuser detection in a dynamic environment – Part II: Joint user identification and parameter estimation. *IEEE Transactions on Information Theory* 55, 5 (2009), 2365–2374.
- [2] Adegbeniga B Awoseyila, Christos Kasparis, and Barry G Evans. 2009. Robust time-domain timing and frequency synchronization for OFDM systems. *IEEE Transactions on Consumer Electronics* 55, 2 (2009).
- [3] IEEE P802.11 TASK GROUP AX. 2017. IEEE P802.11 Wireless LANs: Specification Framework for TGax. *IEEE Standard in Progress* (2017). Available at: http://www.ieee802.org/11/Reports/tgax_update.htm [Online: accessed on 2017-12-10].
- [4] Ezio Biglieri and Marco Lops. 2007. Multiuser detection in a dynamic environment – Part I: User identification and data detection. *IEEE Transactions on Information theory* 53, 9 (2007), 3158–3170.
- [5] Eric Blossom. 2004. GNU radio: Tools for exploring the radio frequency spectrum. *Linux journal* 2004, 122 (2004), 4.
- [6] Panagiotis Botsinis, Dimitrios Alanis, Zunaira Babar, Soon Xin Ng, and Lajos Hanzo. 2015. Iterative quantum-assisted multi-user detection for multi-carrier interleaved division multiple access systems. *IEEE Transactions on Communications* 63, 10 (2015), 3713–3727.
- [7] Yang Du, Binhong Dong, Zhi Chen, Xiaodong Wang, Zeyuan Liu, Pengyu Gao, and Shaoqian Li. 2017. Efficient multi-user detection for uplink grant-free NOMA: Prior-information aided adaptive compressive sensing perspective. *IEEE Journal on Selected Areas in Communications* (2017).
- [8] Adriana B Flores, Sadia Quadri, and Edward W Knightly. 2016. A scalable multi-user uplink for Wi-Fi. In *NSDI*. 179–191.
- [9] Andrea Goldsmith, Syed Ali Jafar, Nihar Jindal, and Sriram Vishwanath. 2003. Capacity limits of MIMO channels. *IEEE Journal on selected areas in Communications* 21, 5 (2003), 684–702.
- [10] Roger Pierre Fabris Hoefel. 2016. IEEE 802.11 ax: A study on techniques to mitigate the frequency offset in the uplink multi-user MIMO. In *Communications (LATINCOM), 2016 8th IEEE Latin-American Conference on*. IEEE, 1–6.
- [11] IEEE 802.11ac. 2014. IEEE Standard for Information technology Local and metropolitan area networks – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput. *IEEE Standards 802.11ac* (2014).
- [12] National Instrument. 2017. Introduction to 802.11ax High-Efficiency Wireless. *White Paper* (2017). Available at: <http://www.ni.com/white-paper/53150/en>.
- [13] Yohan Kim, Sungyoon Cho, and Dong Ku Kim. 2007. Low complexity antenna selection based MIMO scheduling algorithms for uplink multiuser MIMO/FDD system. In *IEEE Vehicular Technology Conference (VTC-Spring)*. 1663–1667.
- [14] Ruizhi Liao, Boris Bellalta, Miquel Oliver, and Zhisheng Niu. 2016. MU-MIMO MAC protocols for wireless local area networks: A survey. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 162–183.
- [15] Kate Ching-Ju Lin, Shyamnath Gollakota, and Dina Katabi. 2011. Random access heterogeneous MIMO networks. In *ACM SIGCOMM*, Vol. 41. 146–157.
- [16] Guowang Miao. 2013. Energy-efficient uplink multi-user MIMO. *IEEE Transactions on wireless communications* 12, 5 (2013), 2302–2313.
- [17] Chester Sungchung Park, Y-P Eric Wang, George Jongren, and David Hammarwall. 2011. Evolution of uplink MIMO for LTE-advanced. *IEEE Communications Magazine* 49, 2 (2011).
- [18] Ettus Research. 2017. USRP N210. www.ettus.com/product/details/UN210-KIT [Online: accessed on 2017-03-08] (2017).
- [19] Stefania Sesia, Matthew Baker, and Issam Toufik. 2011. *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons.
- [20] Arkady Moley Shteiman. 2002. Multi-user detection. US Patent US20040248515.
- [21] Kun Tan, He Liu, Ji Fang, Wei Wang, Jiansong Zhang, Mi Chen, and Geoffrey M Voelker. 2009. SAM: Enabling practical spatial multiple access in wireless LAN. In *ACM MobiCom*. 49–60.
- [22] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.
- [23] Sergio Verdu. 1998. *Multiuser detection*. Cambridge university press.
- [24] Hua Wang, Hung Nguyen, Claudio Rosa, and Klaus Pedersen. 2012. Uplink multi-cluster scheduling with MU-MIMO for LTE-advanced with carrier aggregation. In *IEEE Wireless Communications and Networking Conference (WCNC)*. 1202–1206.
- [25] Sen Wang, Fei Wang, Yafeng Wang, and Dacheng Yang. 2011. A novel scheduling scheme based on MU-MIMO in TD-LTE uplink. In *IEEE Wireless Communications and Networking Conference (WCNC)*. 915–919.
- [26] Michael Wu, Bei Yin, Guohui Wang, Chris Dick, Joseph R Cavallaro, and Christoph Studer. 2014. Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations. *IEEE Journal of Selected Topics in Signal Processing* 8, 5 (2014), 916–929.
- [27] Yik-Chung Wu, Kun-Wah Yip, Tung-Sang Ng, and Erchin Serpedin. 2005. Maximum-likelihood symbol synchronization for IEEE 802.11 a WLANs in unknown frequency-selective fading channels. *IEEE Transactions on Wireless Communications* 4, 6 (2005), 2751–2763.
- [28] Huacheng Zeng. 2017. Demo: Uplink MU-MIMO in Asynchronous Wireless Networks. (2017). Available at: <http://www.ece.louisville.edu/hzeng/mumimo.html> [Online: accessed on 2017-12-12].
- [29] Jiankang Zhang, Sheng Chen, Xiaomin Mu, and Lajos Hanzo. 2012. Turbo multi-user detection for OFDM/SDMA systems relying on differential evolution aided iterative channel estimation. *IEEE Transactions on Communications* 60, 6 (2012), 1621–1633.
- [30] Yangyang Zhang, Chunlin Ji, Yi Liu, Wasim Q Malik, Dominic C O'Brien, and David J Edwards. 2008. A low complexity user scheduling algorithm for uplink multiuser MIMO systems. *IEEE Transactions on Wireless Communications* 7, 7 (2008).
- [31] Ying Jun Zhang. 2010. Multi-round contention in wireless LANs with multipacket reception. *IEEE Transactions on Wireless Communications* 9, 4 (2010).
- [32] Anfu Zhou, Teng Wei, Xinyu Zhang, Min Liu, and Zhongcheng Li. 2015. Signpost: Scalable MU-MIMO signaling with zero CSI feedback. In *Proc. of ACM MobiHoc*. 327–336.