

# Practical MU-MIMO User Selection on 802.11ac Commodity Networks

Sanjib Sur<sup>1\*</sup>, Ioannis Pefkianakis<sup>2</sup>, Xinyu Zhang<sup>1</sup>, Kyu-Han Kim<sup>2</sup>

University of Wisconsin-Madison<sup>1</sup>, Hewlett Packard Labs<sup>2</sup>

sur2@wisc.edu, ioannis.pefkianakis@hpe.com, xyzhang@ece.wisc.edu, kyu-han.kim@hpe.com

## ABSTRACT

Multi-User MIMO, the hallmark of IEEE 802.11ac and the upcoming 802.11ax, promises significant throughput gains by supporting multiple concurrent data streams to a group of users. However, identifying the best-throughput MU-MIMO groups in commodity 802.11ac networks poses three major challenges: *a*) Commodity 802.11ac users do not provide *full CSI* feedback, which has been widely used for MU-MIMO grouping. *b*) Heterogeneous channel bandwidth users limit grouping opportunities. *c*) Limited-resource on APs cannot support computationally and memory expensive operations, required by existing algorithms. Hence, state-of-the-art designs are either not portable in 802.11ac APs, or perform poorly, as shown by our testbed experiments. In this paper, we design and implement *MUSE*, a lightweight user grouping algorithm, which addresses the above challenges. Our experiments with commodity 802.11ac testbeds show *MUSE* can achieve high throughput gains over existing designs.

## Categories and Subject Descriptors

C.2.2 [Computer Systems Organization]: Computer-Communications Networks

## Keywords

Multi-User MIMO, User selection, IEEE 802.11ac

## 1. INTRODUCTION

Multi-User MIMO (MU-MIMO) technology uses precoding (beamforming) to support multiple, concurrent data streams from an Access Point (AP) to a group of users. The resulting theoretical capacity grows proportionally with the number of antennas at the AP. Owing to such advantage, MU-MIMO has been embraced by the latest wireless LAN standard IEEE 802.11ac to realize Gbps downlink. It is also considered as a key high-speed feature for the upcoming IEEE

\*Sanjib Sur completed this work during his internship at Hewlett Packard Labs, Palo Alto.

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

*MobiCom'16*, October 03–07, 2016, New York City, NY, USA.

© 2016 ACM. ISBN 978-1-4503-4226-1/16/10 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2973750.2973758>.

802.11ax [1] and 5G wireless networks [2]. To materialize the huge potential, in addition to precoding, an MU-MIMO AP must select a group of users, whose instantaneous wireless channels are orthogonal and consequently, concurrent transmissions do not cause inter-user interference. Although the design of MU-MIMO user selection has been well established in wireless communication theory, there are 3 major practical challenges for designing user selection in commodity APs.

*(a) Limited feedback:* State of the art algorithms leverage users' Channel State Information (CSI) feedback, to identify the best-throughput user groups [3–7]. However, *full CSI* feedback results in large overheads, which may even nullify the MU-MIMO gains [8]. To overcome this limitation, 802.11ac standard supports only a compressed form of CSI (named *V* matrix) that directly specifies how the AP should precode and de-correlate data across multiple users.

In the absence of users' *full CSI*, legacy algorithms in commodity 802.11ac APs rely on MAC-layer feedback, specifically, Packet-Error-Rate (PER) and PHY rate statistics (MCS), to identify MU-MIMO groups. Specifically, they assign users with similar profiles (*e.g.*, same channel bandwidth and throughput dynamics) to the same MU-MIMO group. They break the group if it suffers from high PER. Despite its simplicity, such a “trial and error” approach can lead to significant performance degradation. Our experiments with commodity testbeds show that Single-User MIMO (where AP serves one user at a time) achieves up to 72% throughput gains over legacy MU-MIMO, for 25% of the experimental cases. The root cause is that PER-based algorithms cannot capture the users' channel correlation and often form groups with high inter-user interference, which raises PER and drops performance. High PER further degrades other MAC/PHY operations such as rate adaptation (RA).

*(b) Heterogenous bandwidth users:* Due to hardware capability and external interferences, not all 802.11ac users can support the same channel bandwidth. However, MU-MIMO users with different channel bandwidths cannot be grouped together, since an AP can only transmit on a single center frequency and channel bandwidth at a time. Hence, heterogeneous channel bandwidth configurations can limit MU-MIMO grouping opportunities and lead to significant throughput degradation. Whereas an AP can force all users to the lowest available channel bandwidth to maximize the likelihood of MU-MIMO grouping, this may sacrifice certain users' channel bandwidth utilization. A proper tradeoff must be made to maximize aggregate users' throughput.

(c) *Limited-resource APs*: Legacy user selection (and other core MAC-layer functionalities) are typically implemented in the firmware of WLAN system in commodity APs to achieve high performance. However, 802.11 chipset vendors curtail memory and CPU capability of their chipsets to reduce costs. Hence, existing MU-MIMO protocols [9–11] that require complex mathematical and memory intensive operations, are not portable to such platforms. Since AP chipset capability has not been significantly changed over the past 8 years (from the advent of 802.11n - *e.g.* [12]), we expect that such system factors will remain a key constraint for future implementations, and must be properly addressed by MU-MIMO designs.

In this paper, we propose a new Mu-mimo User SElection (*MUSE*) design for 802.11ac commodity networks, which addresses the above challenges. *MUSE* leverages 802.11ac’s limited channel feedback to identify the best-throughput MU-MIMO groups and bandwidth configuration. It is able to capture inter-user interference, by computing the  $V$  matrix correlation among users, which acts as proxy of their channel correlation. This leads to a new SINR metric that allows the AP to gauge a user’s potential throughput prior to it joining an MU-MIMO group. Our experiments show that the approximation error of our SINR metric compared to *full CSI* SINR estimator is typically less than 0.5 dB.

Further, *MUSE* is able to boost MU-MIMO gains for heterogeneous (in terms of bandwidth) 802.11ac networks, by optimizing users’ bandwidth configurations, to increase grouping opportunities. Based on the channel information of current bandwidth, it infers the SINR of alternative bandwidths with no additional sounding overhead, and then selects the group-bandwidth combinations with highest throughput. Its inference model is based on our observation that,  $V$  matrix (and hence channel) correlation of the users in an MU-MIMO group is similar across different bandwidths, while there is around 3 dB power gain upon halving the channel bandwidth. To tame the computational cost in searching for the best combination, *MUSE* uses an *informed greedy user selection*, which is able to prune in advance from the search space, those groups with suboptimal throughput.

To enable real-time execution of user selection in resource-constraint APs, we introduce a new DMA-engine-based kernel-firmware communication architecture that allows key MU-MIMO functionalities (CSI processing, SINR estimation, etc.) to be efficiently migrated to the AP’s kernel space running the relatively powerful general-purpose processor with larger memory. This architecture can be reused by all 802.11ac-compatible, real-time MU-MIMO protocols that rely on CSI processing. We validate the efficiency of *MUSE* design in a network comprised of commodity 802.11ac APs and smartphones. For comparison, we also implement PUMA [13] user selection, which does not require CSI and can be ported to 802.11ac APs. Our results show up to  $2\times$  and  $4\times$  aggregated and per-user throughput gains respectively, over the legacy algorithm and PUMA, in controlled settings. In field trials, per-user throughput gains can be up to 61%.

In summary, our contributions are the following:

(1) We conduct a measurement study of commodity 802.11ac MU-MIMO networks, and identify the limitations of legacy MU-MIMO user selection designs (Sec. 3). To the best of our knowledge, this is the first work that characterizes MU-MIMO performance in 802.11ac commodity testbeds.

(2) We design *MUSE*, a lightweight 802.11ac-compliant system that employs a new SINR metric for optimized user selection, taking into account the limited CSI feedback constraint and the heterogeneous bandwidth among users (Sec. 4).

(3) We design and implement a firmware-kernel communication architecture that enables *MUSE* to run on resource-constraint, commodity 802.11ac APs (Sec. 5). This is the first such implementation, since existing MU-MIMO user selection protocols are mainly evaluated through analysis, simulations, or using software radios where all processing is done in PCs.

(4) We evaluate *MUSE*’s performance, in various static, dynamic controlled settings and through larger-scale, realistic field trials with 802.11ac smartphone users (Sec. 6).

## 2. IEEE 802.11AC BACKGROUND

IEEE 802.11ac operates on 5GHz, and supports denser modulation (up to 256-QAM) and faster MIMO (up to 8 streams and 6.9 Gbps rates) compared to its predecessor 802.11n. It also supports frame aggregation where several MAC Protocol Data Units (MPDUs) are aggregated into an A-MPDU to amortize protocol overheads. The key differentiator over 802.11n is the 802.11ac MU-MIMO feature, which uses beamforming to support concurrent downlink data streams from an AP to a group of users.

To support MU-MIMO beamforming, an 802.11ac AP needs to first follow a sounding protocol [14] to probe users and collect a *VHT Compressed Beamforming Feedback (CBF)* from them. The CBF is represented by  $V$ , essentially a steering matrix that specifies how the AP should decorrelate the transmission data to multiple users. Suppose  $H$  is the channel matrix measured at the user’s side from sounding packet. The user calculates  $V$  by applying Singular Value Decomposition (SVD) on  $H$ :  $H = UDV^H$ . To reduce the overhead in feedback transmission,  $V$  is further compressed through Givens Rotation [14, 15], quantization and grouped over multiple subcarriers [14], which can reduce the overhead by 129.1% to 268.4% compared to *full CSI*  $H$  depending on the number of AP’s antennas. Given  $V$ , the AP precodes transmission data  $x$  as  $u = Vx$  following the Eigen-subspace beamforming [16]. Upon receiving the precoded data, each user applies the matrix  $U$  to extract its own data, treating others’ data as interference:

$$U^H y = U^H (Hu + n) = U^H (UDV^H) Vx + \tilde{n} = Dx + \tilde{n} \quad (1)$$

Apart from CBF, 802.11ac users provide a *MU Exclusive Beamforming report* to AP, which is only available in MU-MIMO mode, and carries per-subcarrier delta SNRs along with an average (across subcarriers) SNR. This feedback can be used by AP to estimate the SNR of different subcarriers. However, this SNR is calculated from the sounding packet and does not capture inter-user interference. Note that, the complete channel matrix  $H$  (CSI) is a  $N_r \times N_t$  matrix (where  $N_r$  and  $N_t$  are the number of receive and transmit antennas, respectively) and it cannot be computed from  $V$  and SNR.

An 802.11ac AP decides upon a set of users to transmit data concurrently through a *user selection* algorithm that precedes the MU-MIMO sounding and beamforming. User selection algorithm is not specified by the 802.11ac standard and it is AP vendor’s implementation specific. Each user within an MU-MIMO group can operate with independent PHY rate, identified by a *rate adaptation* algorithm.

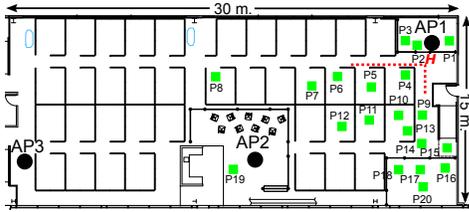


Figure 1: Experimental floorplan. Spots AP1-AP3 and P1-P20 mark the locations where APs and users are placed.

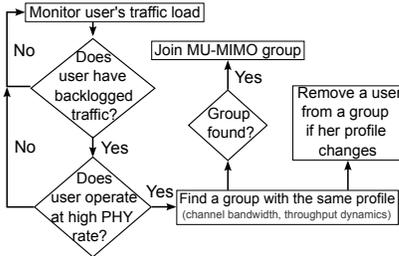


Figure 2: Overview of Legacy-US.

802.11ac supports 20, 40, 80 MHz channel bandwidths, and an optional 160 MHz bandwidth. An 802.11ac device can use a 20 MHz sub-channel only if it is not occupied by another transmission. To negotiate a higher channel bandwidth, an 802.11ac AP sends an RTS to the receiver on each of the sub-channels (e.g., 4 RTS for 80MHz). The receiver responds with one CTS for each unoccupied sub-channel. RTS/CTS negotiation is not required on a per-packet basis and its implementation is again vendor specific.

### 3. 802.11AC MU-MIMO PERFORMANCE

In this section, we conduct extensive experiments to understand the working mechanisms and limitations of legacy 802.11ac user selection.

#### 3.1 Platform and Methodology

Our experiments use commodity AP boards, equipped with a 4×4 MU-MIMO-capable 802.11ac 5 GHz radio. The 802.11ac radio supports up to 80 MHz channel bandwidth and up to 256-QAM modulation level, with 1733.3 Mbps peak PHY rate. It has 4 antennas, but only supports up to 3 data streams (users) in MU-MIMO mode. MU-MIMO user selection and rate adaptation are implemented in the board's firmware, and the source code is available for our modifications. Our experiments adopt Xiaomi Mi 4i smartphones [17] as users. Xiaomi Mi 4i has a 802.11ac wave-2 chipset, with one receiving antenna.

We conduct our experiments in an office building (floor plan shown in Fig. 1), using both controlled experiments (interference-free, without human mobility) and field trials.

#### 3.2 Overview of Deployed 802.11ac Designs

**User selection:** An overview of our platform's MU-MIMO user selection algorithm (named as Legacy-US for Legacy User Selection) is shown in Fig. 2. A user can join an MU-MIMO group, only if it has sufficient backlogged traffic and operates at a high PHY rate. The latter design choice seeks to prevent users operating at low rates, from further dropping their rates in an MU-MIMO setting, where transmit power is shared among users. An eligible user can then join an MU-MIMO group of users with similar profiles. The *pro-*

*file* includes channel bandwidth and temporal throughput dynamics. First, only users with the same channel bandwidth can be grouped together. This is because an AP can only transmit using a single center frequency and bandwidth at a time. Moreover, users of different bandwidth may have different interference profiles, and hence the AP cannot transmit data to all of them using the highest channel bandwidth. Second, Legacy-US groups users with similar throughput gradient (i.e., throughput changes similarly over time). If the group results in high PER, the users' throughput profiles will change and no longer match the group's profile. Consequently, users will be removed from the group. **Rate adaptation (RA):** The RA algorithm seeks to identify the best-throughput PHY rate (MCS, channel bandwidth and number of spatial streams), at runtime. Our platform uses a variant of *Minstrel* [18], which maintains per-user PER statistics for each rate, updated upon the reception of ACK frames. Minstrel uses PER to estimate the throughput under each bit-rate choice.

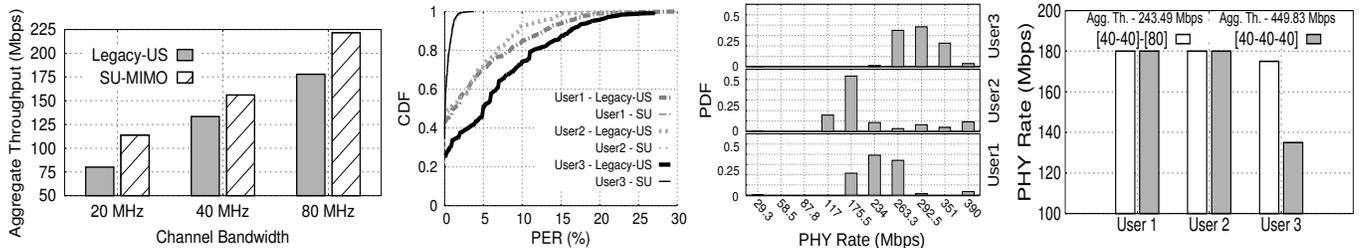
The above algorithms are representative of what is implemented in commodity APs and proposed by research studies. PER-based RA has been widely adopted by large 802.11 vendors such as Qualcomm and Broadcom for its simplicity, and widely studied in the literature [19, 20]. PER-based user selection for MU-MIMO is currently implemented in the commodity APs.

### 3.3 A Case Study

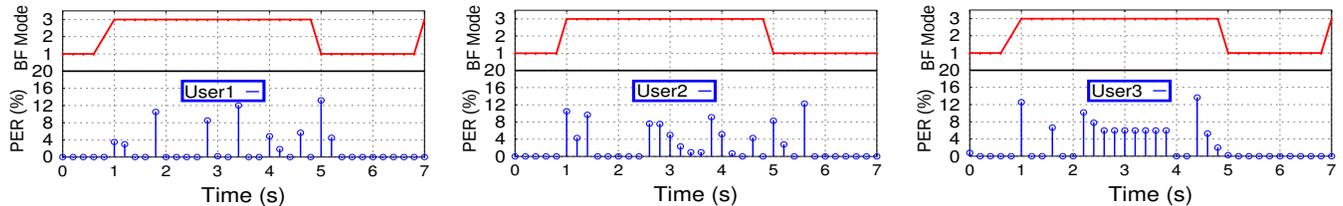
**Legacy MU-MIMO performance diagnosis.** To profile Legacy-US, we start with a controlled setting where an AP transmits back-to-back UDP traffic to 3 static users. Our setting is free of external interference and human mobility. We compare Legacy-US with the *SU-MIMO mode*, where the AP serves one user at a time. Note that Xiaomi phones only have one receiving antenna, so the SU-MIMO mode only supports one data stream. Ideally, we expect that MU-MIMO should have much higher throughput than SU-MIMO, since channel is time-shared in SU-MIMO mode.

Surprisingly, our case-study experiment (Fig. 3(a)) demonstrates that SU-MIMO achieves 16.8% to 42% higher aggregated throughput compared to Legacy-US, for all channel bandwidth settings. We found the poor Legacy-US performance roots in users' higher PER. Fig. 3(b) shows the distribution of the per-user PER averaged over 200ms windows. While SU-MIMO PER is mostly zero and never exceeds 4%, Legacy-US PER can reach up to 30%. Given the external-interference free setting, we attribute Legacy-US's poor performance to the inter-user interference, which happens when users with correlated channels are grouped together [9].

The high PER changes the users' throughput gradient profiles and consequently breaks the MU-MIMO group (cf. Fig. 2). We call this phenomenon *group thrashing*. We illustrate group thrashing in Fig. 4, which shows the variation between beamforming (BF) mode and the PER, across 7 seconds. The BF mode for user  $k$  is defined as the number of users in user  $k$ 's MU-MIMO group. BF mode equals to 1 when  $k$  is the only user in a group (i.e., SU-MIMO mode). We can see that during the 0s-1s time window, all users start with SU-MIMO and have similar PER and thus throughput dynamics. So Legacy-US will force them to join the same MU-MIMO group. Afterwards (1s-5s), the MU-MIMO PER due to inter-user interference can exceed 12% for all the users. This will make the group break at the 5<sup>th</sup>



**Figure 3:** Legacy-US performance in our case study setting. (a) Legacy-US and SU-MIMO aggregated throughput. (b) Legacy-US and SU-MIMO PER (80 MHz). (c) Legacy-US rate distribution (80 MHz). (d) Benefit of bandwidth adaptation.



**Figure 4:** PER and beamforming mode variation for our case study users (80 MHz).

second. Even after group thrashing, PER may remain high until the rate adaptation (RA) protocol re-identifies the best rate (*e.g.*, 5.6s for user 2). We have observed that such group thrashing happens recurrently over a long term. As a result, the average MU-MIMO network throughput becomes lower than SU-MIMO.

**Impact on rate adaptation:** Inter-user interference and group thrashing negatively impact the RA algorithm as well. To gauge the RA algorithm, we first measure the best-throughput fixed-rate when users operate in SU-MIMO, and when all users are forced to join an MU-MIMO group (Fig. 5(a)). Fig. 3(c) plots a histogram of the PHY rates chosen under Legacy-US, which span many levels other than the best-throughput ones. For example, whereas the best fixed rate for user 3 is MCS 9 (390 Mbps) in SU-MIMO mode and MCS 6 (263.3 Mbps) in MU-MIMO, 62% of Legacy-US transmissions occur at different rates than the best ones. The poor adaptation roots in group thrashing. Since a user’s best PHY rate can be different in MU-MIMO and SU-MIMO modes, every time a user joins/leaves a group, RA needs time to converge to the new best rate. Even an oracle RA algorithm that can identify the instantaneous best rate cannot compensate for poor MU-MIMO user selection. It can only react by switching to lower a MCS to cope with the inter-user interference.

**Impact of heterogeneous channel bandwidths:** Users with different channel bandwidths cannot be grouped together (cf. Sec. 2 and Sec. 3.2), which limits MU-MIMO grouping opportunities. To illustrate the impact of this constraint, we repeat our case study, in a setting where two users operate at 40 MHz and one at 80 MHz. Legacy-US will group the two 40 MHz users, leaving the 80 MHz user at SU-MIMO mode. Interestingly, forcing all users to 40 MHz and putting them in one MU-MIMO group leads to 85% higher aggregated throughput over Legacy-US. Although, reducing an 80 MHz user to 40 MHz may reduce its own PHY rate (as shown in Fig. 3(d))<sup>1</sup>, simultaneous data stream transmission to the users of an MU-MIMO groups can compensate for such loss.

<sup>1</sup>Interestingly, it did not reduce the PHY rates for User 1 & 2 in this case, since the power reductions due to more streams were not sufficient enough to change their operating MCSs.

**Summary:** Our results show that the MAC-layer feedback based MU-MIMO user selection designs in 802.11ac devices perform poorly. Specifically, Legacy-US cannot properly identify user groups with orthogonal channels, which results in inter-user interference and high PER. It leads to group thrashing and sub-optimal rate selection. Simply optimizing RA cannot compensate for erroneous MU-MIMO user selection either. Further performance degradation results from the disjoint user selection and channel bandwidth adaptation. *Our findings make the case for a user selection framework that can effectively account for the user’s channel orthogonality and bandwidth heterogeneity, while efficient enough to be executable on commodity 802.11ac APs.*

### 3.4 Performance Over Multiple Locations

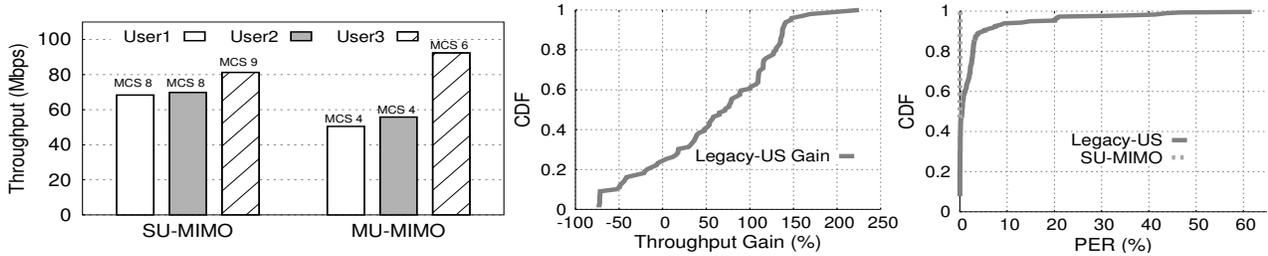
We verify the generality of our findings, by repeating the case study experimental setting to more than 100 locations with various channel bandwidths. Our goal is to both understand the limitations of Legacy-US in multiple settings, and the MU-MIMO gains when user selection can identify users with uncorrelated channels. Fig. 5(b) shows the distribution of Legacy-US gains over SU-MIMO. We observe that in approximately 25% of the settings, the gain is negative, and SU-MIMO achieves up to 72% higher throughput. Fig. 5(c) shows distribution of users’ PER. Whereas PER is negligible in SU-MIMO, it can exceed 60% for Legacy-US. Interestingly, for certain settings, MU-MIMO has comparable PER and more than 2× throughput gain over SU-MIMO. These represent cases where Legacy-US happen to pick optimal user groups, and hints to the importance of grouping users with uncorrelated channels.

## 4. MUSE DESIGN

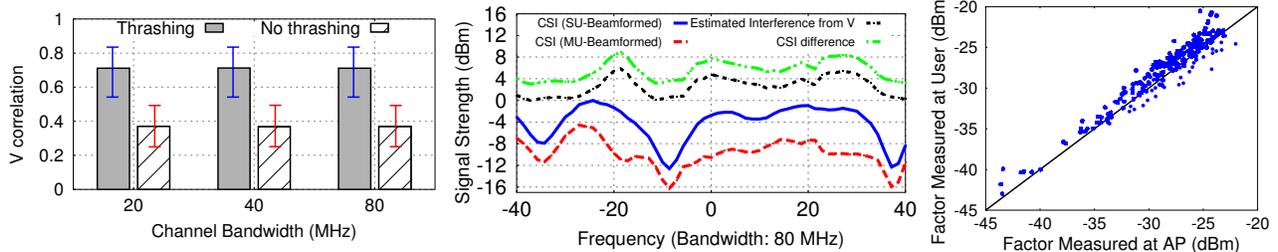
We next present *MUSE*, which seeks to identify the throughput-maximizing MU-MIMO group along with the PHY rate and channel bandwidth settings.

### 4.1 MUSE SINR Estimation

A major challenge towards designing *MUSE* is to estimate the inter-user interference and compute the SINR of a user before putting it into an MU-MIMO group. Then, SINR can be used for identifying best-throughput MU-MIMO groups.



**Figure 5:** (a) MU- vs. SU- performance under fixed rate in case study. (b) Legacy-US throughput gain over SU-MIMO in multiple static settings. (c) Legacy-US PER in multiple static settings.



**Figure 6:** Evaluating the factors of  $\mu\text{SINR}$ . (a)  $V$  correlation for thrashing/no thrashing cases. (b) An example of interference estimation using  $V$  matrix. (c) Factor  $\|D_k\|^2/N$  at the user and AP sides.

While the theory for MU-MIMO SINR estimation [9, 21] is well-established, a fundamental question remains open in practice: How can SINR be estimated in the absence of users' *full CSI* feedback in practical 802.11ac networks?

In *MUSE*, we model inter-user interference and predict SINR, by manipulating the CBF matrix  $V$  from users. Recall that, users'  $V$  matrix specifies how the AP should decorrelate the transmission data across users of the same group. Therefore,  $V$  matrix intuitively should reflect users' channel correlation. We experimentally validate our intuition, by measuring the correlation of  $V$  matrices between pair of users at subcarrier  $s$  as [22]:

$$\rho(i, j) = \frac{\sum_s \|V_i(s)V_j^H(s)\|}{\sqrt{\sum_s \|V_i(s)\|^2} \sqrt{\sum_s \|V_j(s)\|^2}} \quad (2)$$

Fig. 6(a) shows the average  $V$  correlations among users of MU-MIMO groups, from all our experimental settings. The error bars show the max. and min. correlation. We differentiate the cases where MU-MIMO performs better than SU-MIMO (no thrashing), and where it performs poorly (thrashing). We observe  $V$  correlation to be almost  $2\times$  higher in MU-MIMO thrashing cases, independently of channel bandwidth. *Our results corroborate that  $V$  correlation is a proxy of the inter-user interference.* In what follows, we formally introduce the model to estimate the inter-user interference leveraging this  $V$  matrix.

**Interference model:** An 802.11ac AP precodes the transmission data ( $x$ ), following the eigen-subspace beamforming as  $Vx$  (Sec. 2). Let  $y_k$  be the received signal at user  $k$ . From Eq. (1), when user  $k$  applies the matrix  $U$  to its received signal, we have:

$$\begin{aligned} U_k^H y_k &= U_k^H H_k Vx + U_k^H n_k \\ &= U_k^H (U_k D_k V_k^H) V_k x_k + U_k^H (U_k D_k V_k^H) \sum_{j \neq k} V_j x_j + \widehat{n}_k \\ &= \underbrace{D_k x_k}_{\text{Signal}} + \underbrace{D_k \sum_{j \neq k} V_k^H V_j x_j}_{\text{Interference}} + \underbrace{\widehat{n}_k}_{\text{Noise}} \end{aligned} \quad (3)$$

We see that the interference term is a function of the  $V$  correlation between users within the MU-MIMO group. It also

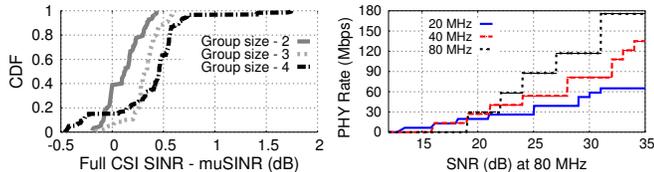
depends on the  $D_k$ , the singular-value matrix corresponding to user  $k$ 's channel. To validate our interference model, we use our testbed to collect per-subcarrier CSI traces at the user-side, when a user is served in SU-MIMO mode, and when the same user is a part of an MU-MIMO group with another user. We then measure the difference between the user's SU-MIMO CSI and MU-MIMO CSI, which reflects both the inter-user interference and transmit power drop caused by MU-MIMO. Fig. 6(b) shows that the inter-user interference estimated from our model in Eq. (3) can consistently follow the CSI difference. Specifically, the CSI difference (which varies across subcarriers), is approximately 3 dB when inter-user interference is negligible, since the transmit power in MU-MIMO is distributed to two users in our example. The CSI difference can rise to 8 dB for particular subcarriers when inter-user interference is high, as captured by our model. Note that, the model in Eq. (3) does not capture power split across multiple users and thus there exist a constant difference of 3 dB across subcarriers between the estimation and measured CSI difference (Fig. 6(b)).

**$\mu\text{SINR}$  metric:** *MUSE* SINR ( $\mu\text{SINR}$ ) metric is a function of signal strength  $P$  towards a user  $k$ , inter-user interference  $I^2$  and noise floor  $N$ :  $\text{SINR} = P/(I + N)$ . Based on Eq. (3), we can model signal strength as:  $P = \mathbb{E}\{D_k x_k\{D_k x_k\}^H\} = (1/K)\|D_k\|^2$  (where  $\mathbb{E}[x_k x_k^H] = (1/K)$ , as the AP's transmission power is equally split among  $K$  users). Further, inter-user interference power is:

$$\begin{aligned} I &= \mathbb{E}\{D_k \sum_{j \neq k} V_k^H V_j x_j \{D_k \sum_{j \neq k} V_k^H V_j x_j\}^H\} \\ &= \mathbb{E}[D_k D_k^H] \mathbb{E}\{\{\sum_{j \neq k} V_k^H V_j x_j\} \{\sum_{j \neq k} V_k^H V_j x_j\}^H\} \\ &= (1/K)\|D_k\|^2 \sum_{j \neq k} \|V_k^H V_j\|^2 \end{aligned} \quad (4)$$

Note that in Eq. (4),  $D$  and  $V$  are different random variables and can be considered uncorrelated. Then the AP estimates the  $\mu\text{SINR}$  of user  $k$  as:

<sup>2</sup>*MUSE* omits interference from other Wi-Fi networks, which is typically small, since the 802.11 MAC precludes concurrent transmission from adjacent APs.



**Figure 7:** (a) *muSINR vs. full CSI SINR (80 MHz)*. (b) *PHY rate comparison at different channel bandwidths*.

$$\begin{aligned}
 \text{SINR} &= \frac{\frac{1}{K} \|D_k\|^2}{N + \frac{1}{K} \|D_k\|^2 \sum_{j \neq k} \|V_k^H V_j\|^2} \\
 &= \frac{1/K}{(1/\|D_k\|^2/N) + \frac{1}{K} \sum_{j \neq k} \|V_k^H V_j\|^2}
 \end{aligned} \quad (5)$$

muSINR estimation requires matrices  $V$ ,  $D$  and noise floor  $N$ . While  $V$  is directly fed back by the user,  $D$  and  $N$  are unknowns. We compute factor  $\|D_k\|^2/N$  from average SNR ( $\overline{\text{SNR}}_k$ , averaged over all sub-carriers) and delta SNR ( $\Delta \text{SNR}_k$ , difference of per-subcarrier SNR and  $\overline{\text{SNR}}_k$ ) provided by the CBF and MU Exclusive Beamforming reports, as [14]:

$$\|D_k\|^2/N = 10^{\frac{\overline{\text{SNR}}_k + \Delta \text{SNR}_k}{10}} \quad (6)$$

**muSINR accuracy:** We conduct experiments to evaluate the accuracy of the muSINR metric, in comparison with an oracle estimator that has the *full CSI* matrix ( $H$ ). To this end, we first collect channel traces from 10 randomly located users associated to one AP. Then we compute muSINR and *full CSI* SINR by randomly grouping users, in group sizes ranging from 2 to 4 (maximum size supported by IEEE 802.11ac). Fig. 7(a) shows the distribution of the SINR differences. We observe that muSINR differs from the oracle SINR by a small median error of 0.2 dB to 0.4 dB across group sizes, despite its use of the CBF  $V$ . The muSINR error never exceeds 0.5 dB and 0.6 dB for groups with 2 and 3 users, respectively. Even for a group of 4 users, muSINR has less than 1 dB error for 98% of the samples. Since the 802.11ac PHY rate options' SINR thresholds are separated by at least 1 dB [14], this small error is unlikely to affect *MUSE* PHY rate adaptation and user selection.

## 4.2 Estimating $D$ and $N$

The computation of muSINR requires the factors  $D$  and  $N$  (cf. Eq. (6)). These factors can be extracted from the MU Exclusive Beamforming feedback, but are unavailable before a user is selected into an MU-MIMO group. To circumvent this barrier, *MUSE* leverages channel strength reciprocity, a property widely used to evaluate both signal strength [23] and noise [24] in commodity 802.11 devices<sup>3</sup>. First, it estimates  $D$  by applying SVD to CSI ( $H$ ) measured at the AP side, collected from uplink control and data frames. Since AP and users typically have different transmit power configurations, *MUSE* calibrates  $D_k$  to account for the power factor. Specifically, it multiplies the factor  $\|D_k\|^2/N$  with  $10^{\frac{P_{AP} - P_{user}}{10}}$ , which is the transmit power difference at AP and user sides.  $P_{user}$  is sent from users to APs through the 802.11 Event Report frames during association [14]. Finally, *MUSE* estimates the noise  $N$  using EVM (Error Vector Magnitude), a statistic that is originally used for rate adaptation, and available for every received frame's pilot subcarriers.

<sup>3</sup>Channel strength reciprocity does not imply channel phase reciprocity, on which the inter-user interference depends.

We conduct experiments to verify the accuracy of  $D$  and  $N$  estimation. Fig. 6(c) compares the factor  $\|D_k\|^2/N$  at AP and user sides, estimated from similar channel traces as above. We observe that the  $\|D_k\|^2/N$  factor estimated by the AP matches close with the ground truth at user side. Even though the difference may occasionally reach up to 4 dB, the impact on muSINR estimation is much smaller because of the inter-user interference term (cf. Eq. (5)).

Note that, MU-MIMO user selection requires the CBF matrix  $V$  from users, which is still available to the AP prior to group formation — the AP can beamform to the individual users and collect CBF matrix  $V$  while communicating in SU-MIMO mode. *MUSE* leverages such initial feedback for selecting users in groups, which we detail below.

## 4.3 Bandwidth-Aware User Selection

Existing MU-MIMO user selection protocols are oblivious of the heterogeneous channel bandwidths among 802.11ac users [3, 5–7, 9, 13]. However, our experiments in Sec. 3.3 show that such an approach can limit MU-MIMO grouping opportunities, since users of different channel bandwidths cannot be grouped together. In *MUSE*, we design a joint user selection and bandwidth adaptation mechanism to maximize MU-MIMO gains. The key challenge for such a design is to infer muSINR (and hence MU-MIMO performance) at different bandwidths and to identify the best groups with additional protocol overhead and computational cost. We next present how *MUSE* addresses these challenges.

### 4.3.1 Low Cost muSINR and Throughput Estimation

*MUSE* needs to compute users' muSINRs for each available channel bandwidth, to identify the best-throughput bandwidth and MU-MIMO group settings. A naive solution would be to sound each user at every bandwidth, which results in significant overheads.

**muSINR inference:** *MUSE* can infer the  $D$ ,  $N$  and  $V$  correlation factors, which are required for the computation of muSINR (cf. Eq. (5)) for every bandwidth option with no additional sounding, based on two key observations.

*First*, our experiments in Fig. 6(a) show that, *the  $V$  matrix correlation among users in an MU-MIMO group is similar across different channel bandwidths*. Intuitively, the channel correlation among users depends on the similarity of the shape of their  $V$ s. However, above a certain bandwidth (e.g., 20 MHz), small changes in the shapes of  $V$ s do not affect the correlation value. We have verified our finding in multiple settings. Hence, we can approximate the  $V$  correlation for every bandwidth, based on the measured  $V$  correlation of the current bandwidth setting. *Second*, an AP proportionally reduces the power per subcarrier as the bandwidth is increased, to maintain a constant total transmit power. For example, given a total transmit power  $P$ , the AP doubles the transmit power per subcarrier at 40 MHz compared to 80 MHz, which ideally corresponds to a 3 dB gain per subcarrier. Given that for a channel bandwidth  $BW_1$  the signal strength is  $\|D_k\|^2$  (cf. Eq. (5)), then, the signal strength for a new bandwidth  $BW_2$  is  $\alpha \cdot \|D_k\|^2$ , where  $\alpha = BW_1/BW_2$ . From Eq. (5), the per-subcarrier muSINR at different bandwidths can be estimated as:

$$\text{SINR} = \frac{\alpha \cdot \frac{1}{K} \|D_k\|^2}{N + \alpha \cdot \frac{1}{K} \|D_k\|^2 \sum_{j \neq k} \|V_k^H V_j\|^2} \quad (7)$$

---

**Algorithm 1** *Informed Greedy User Selection*

---

```
1: Initialize & sort user set:  $U = \{u_1, u_2, \dots, u_N\}$ . Define  $T_{80}, T_{40}, T_{20}$ 
2: Final group set:  $\phi = \{\}$ . Incomplete group set  $\psi = \{\}$ 
3: for  $bw \in \{80, 40, 20\}$ 
4:   greedy_user_selection( $U, bw, \phi, \psi$ )
5:   Remove the users included in set  $\phi$  from  $U$ . Go to step 3.
6: end for
7: return  $\phi \cup \psi$ 
8: function greedy_user_selection( $U, bw, \phi, \psi$ )
9:   Find the best-throughput group/user  $g \in \psi \cup \{u_i\}$  for  $u_i \in U$ 
10:   Set bandwidth for  $g$  to  $bw$  (if possible) and name it as  $g'$ 
11:   foreach  $u_i \in U$ 
12:     If  $(Th(g' \cup \{u_i\}) > (Th(g') + Th(u_i))/2 \& SINR_{u_i \in g'} > T_{bw} \& bw_{u_i} == bw \& g'$  incomplete)
13:        $g' = g' \cup \{u_i\}$ 
14:     If ( $g'$  is complete)
15:       If  $(Th(g') > Th(g)) \phi = \phi \cup \{g'\}$    else  $\phi = \phi \cup \{g\}$ 
16:     elseif  $(Th(g') > Th(g)) \psi = \psi \cup g'$  else  $\psi = \psi \cup g$ 
17: end function
```

---

Based on the model, *MUSE* infers muSINR in other bandwidths, without any additional sounding.

**RA and throughput model:** The muSINR metric is estimated per OFDM subcarrier. We calculate an *effective muSINR* across all subcarriers using the approach proposed in [24], which has been shown to be robust in frequency-selective multipath environment. *MUSE* uses the effective muSINR to select the best-throughput PHY rate (*i.e.*, MCS), spatial streams and channel bandwidth. It maps the muSINR to a PHY rate based on the 802.11ac rate tables [14].

Given the PHY rate estimation, *MUSE* further estimates the aggregated throughput ( $Thr$ ) of an MU-MIMO group.  $Thr$  is a function of: (1) the PHY rate  $r_k$  for a user  $k$  joining an MU-MIMO group, (2) user's backlogged traffic  $b_k$  and frame size, (3) the protocol overheads  $T_O$  related to sounding and data transmission, and can be defined as:  $Thr = \frac{S_D}{T_D + T_O}$ . The amount of data ( $S_D$ ) to be transmitted at a user  $k$  depends on both the backlogged traffic  $b_k$  and the maximum frame (A-MPDU) size  $S_{ampdu-max,k}$ , which may be user-specific [19].  $S_{ampdu,k}$  equals to  $b_k \cdot S_{ampdu-max,k}$ , and represents the aggregated frame size for a particular user, where  $b_k \in [0, 1]$ . The transmitted data is then  $S_D = \sum_{k=1}^{|m|} S_{ampdu,k}$ , where  $|m|$  is the number of users in an MU-MIMO group set  $m$ . *Our model captures both the traffic in users' queues and their achieved frame aggregation level.* The data transmission time ( $T_D$ ) is modeled as  $T_{ampdu,k} = T_{vht-p} + S_{ampdu,k}/r_k$ , where  $T_{vht-p}$  is the PLCP preamble transmission time, and  $S_{ampdu,k}/r_k$  is the frame transmission time. The total time is:  $T_D = \max_{k \in m} T_{ampdu,k}$ . The protocol overhead ( $T_O$ ) for an MU-MIMO setting is the sum of sounding and ACK overhead. Our AP platform maintains per-user state that includes traffic in users' queues, aggregated frame size, protocol overhead parameters, which are required for the above throughput estimation.

Note that, the muSINR, PHY rate and throughput estimations are all executed by the AP before it selects users for MU-MIMO grouping. We now proceed to describe how *MUSE* forms user groups based on such estimations.

### 4.3.2 User Grouping

*MUSE* seeks to increase the MU-MIMO grouping opportunities by adjusting the users' channel bandwidth to allow

for larger groups. For example, in the case study setting of Fig. 3(d), *MUSE* will lower the 3<sup>rd</sup> user's 80 MHz channel to 40 MHz, to allow for a group of 3 users at 40 MHz. Since, *MUSE* can estimate a group's aggregated throughput across all channel bandwidths using its muSINR inference model, it could use exhaustive search [13] to form the optimal MU-MIMO groups. However, exhaustive search is computationally prohibitive for resource constraint APs, since for  $C$  user-bandwidth pairs, its search space is  $O(C!)$ . The SIEVE user selection [25] uses a branch and bound search approach to reduce complexity. SIEVE has quadratic complexity with the number of users. It uses a factor  $K$  to capture the tradeoff between computational complexity and channel coherence time, and to prune the branch and bound tree. It uses empirical measurements to relate  $K$  with computational time. However, for APs that run multi-process Linux OS, the computational time for a particular process varies significantly with the AP load, and is hard to predict.

*MUSE* adopts an *informed greedy user selection*, which is able to exclude from the search space those user groups with suboptimal throughput, based on the following observation.

**Property 1** *An MU-MIMO group of users at  $BW_1$  provides strictly higher aggregate throughput than the same group at  $BW_2$  (where  $BW_1 > BW_2$ ), only when all the users in the group have an  $SINR_{BW_1} > T$ .*

The above property is derived from the MCS - SNR mappings of different channel bandwidths, as defined in 802.11ac standard [14]. For example, Fig. 7(b) shows that  $BW_1 = 80\text{MHz}$  achieves strictly higher bit-rate than  $BW_2 = 40\text{MHz}$  only when threshold  $T \geq 22$  dB.

From the above property, *MUSE* knows in advance that a *complete* group  $g$  (*i.e.*, a group with no degrees of freedom left) operating at  $BW_i$ , whose users'  $SINR_{BW_i} > T$ , performs better than  $g$  operating at a lower channel bandwidth. Hence, it does not need to evaluate lower bandwidths for  $g$ . Based on this rationale, *MUSE* starts a greedy search from the highest bandwidth (*e.g.*, 80MHz), and considers only the users who can support that bandwidth<sup>4</sup>. *MUSE* first sorts in descending order, the users based on their current throughput and iteratively goes through the list to group the users that provide the highest aggregate throughput with those already selected users. At each iteration, it also ensures that user's SINR is greater than  $T$ , to satisfy the constraint of *Property 1*. The search terminates when the group is complete, or when adding more users to a group results in lower aggregate throughput than serving them in SU-MIMO mode. Greedy search is repeated at lower bandwidths for only the *incomplete* groups, to allow for more grouping opportunities. Alg. 1 shows the steps in detail. Greedy search is less computationally expensive than exhaustive search, with linear complexity to the number of users in the best and average case. *MUSE*'s informed greedy search can reduce computational time up to 67% (given 3 bandwidth options) compared to a simple greedy search.

*MUSE* may lower a user's bandwidth to allow for higher throughput groups. As a positive side-effect, *MUSE* can reduce co-channel interference in 802.11ac networks. Note that changing a user's bandwidth requires RTS/CTS handshake, which happens only during group update, and takes negligible time.

---

<sup>4</sup>Our AP sends RTS periodically to negotiate a user's bandwidth, based on the user's interference profile.

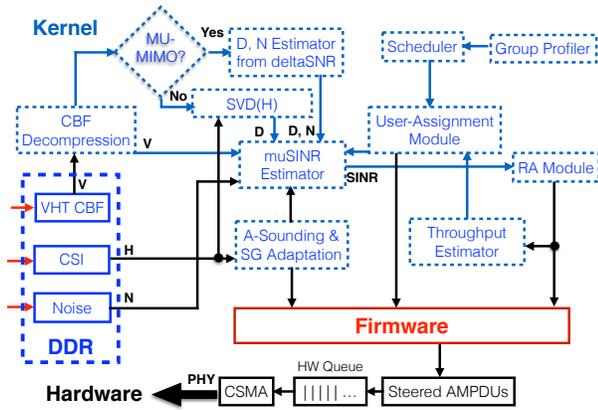


Figure 8: MUSE system architecture.

#### 4.4 Putting Everything Together

An overview of MUSE system architecture is shown in Fig. 8. MUSE leverages a fast DMA (Direct Memory Access) engine to copy the compressed  $V$  matrices, delta SNRs, CSIs and noise estimation from the firmware to the DDR (we elaborate on this engine in Sec. 5). This allows for MUSE core functionality to be implemented in the kernel, which has access to larger memory and more powerful CPU than the firmware. MUSE decompresses  $V$  matrices and feeds them along with delta SNR into the muSINR estimator module. If delta SNRs are not available, it estimates  $D$ ,  $N$  from CSI and EVM measured at the AP, and inputs them to muSINR module (Sec. 4.2). The effective user’s muSINR for a particular group assignment is used by the RA module to select user’s best-throughput PHY rate. Throughput is estimated based on the selected rate and is used by the user selection module to identify best-throughput MU-MIMO grouping along with channel bandwidths. MUSE further estimates coherence bandwidth and time using CSI, to reduce computation costs and to adapt sounding and subcarrier grouping (detailed in Sec. 5). The selected rates (MCS, streams, bandwidth) and MU-MIMO groups are used to program hardware registers through the firmware.

User grouping can work in concert with existing scheduling policies. For example, scheduling may require users with the same traffic profiles and priorities to be scheduled at the same time (to maintain QoE as defined by 802.11e). Group profiling and scheduling modules can work on top of MUSE as shown in Fig. 8, and are independent of its operations.

### 5. IMPLEMENTATION ON COMMODITY ACCESS POINTS

The current off-the-shelf 802.11ac APs employ a dual architecture system. A general-purpose System-on-Chip (*a.k.a.* host) controls the entire AP board, running a variant of Linux OS, while a peripheral WLAN System-on-Chip (*a.k.a.* target) runs the wireless MAC/PHY protocols. MU-MIMO user selection and RA algorithm are implemented in the firmware of WLAN system in the APs, for better performance. However, 802.11 WLAN chipset vendors (such as Qualcomm, Broadcom, Marvell *etc.*) curtail memory and CPU power of their chipsets to mainly reduce costs. Our AP’s firmware has access to only 960 KB static on-chip memory distributed in Data and Instructions RAM, whose 98% is already being utilized by legacy functionalities. MUSE

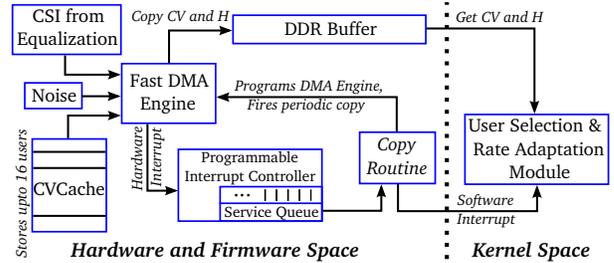


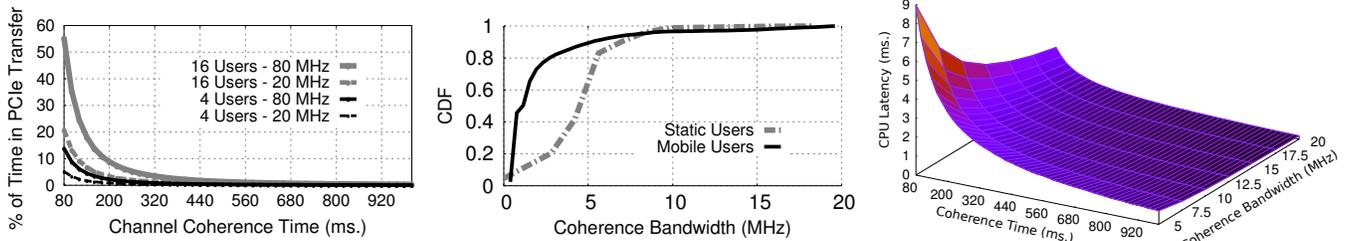
Figure 9: Firmware-kernel interaction architecture.

requires more memory space. For example, for a 4-antenna AP with 16 users operating at 80 MHz, it needs 23 KB for storing the compressed  $V$  matrices and 121 KB for  $H$ . Our AP’s firmware also uses a limited-power 350 MHz CPU. Chipsets capability has not been significantly changed over the past 8 years (*e.g.*, 802.11n AR9331 has similar CPU and 128 KB memory [12]), while the functionalities ported in the firmware are getting more complex and require more memory. Hence, we expect that such system factors will remain a key constraint for future MU-MIMO implementations. We next show how MUSE addresses these challenges.

#### 5.1 Firmware-Kernel Communication Architecture

MUSE overcomes the limited CPU and memory constraints by offloading its core functionality to the comparatively powerful “host” system, which runs the Linux kernel on a dual core 1.4 GHz CPU with a 512 MB DDR3 memory. Fig. 9 illustrates MUSE’s firmware-kernel interaction architecture to realize the offloading. The host communicates with the target system through a 32-bit PCIe bus. MUSE leverages a fast DMA engine to copy the required input for MUSE, from the target system’s on-chip memory to DDR, through the PCIe interface. Specifically, it implements a *Copy Routine* in the firmware that sets up a fast DMA engine to copy  $V$ ,  $H$  and  $N$  from the hardware Compressed  $V$  cache, CSI and Noise registers to DDR address space. The *Copy Routine* periodically (depending on the coherence time update interval) fires the DMA engine to copy per-user  $V$ ,  $H$  and  $N$ . On completion, the DMA generates an interrupt registered by a Programmable Interrupt Controller in an interrupt service queue, which eventually gets serviced. The *Copy Routine* further generates a “soft” interrupt to MUSE modules that run in the kernel of the host. After estimating the best-throughput groups and PHY rates, MUSE sends this information to the firmware, which programs specific hardware registers before MU-MIMO transmissions.

A concern related to our architecture is the PCIe transfer overheads, which could prevent MUSE from being adaptive to channel dynamics. In Fig. 10(a), we measure our AP’s PCIe transfer latencies from on-chip memory to DDR, for varying coherence times and data volume (for a fixed antenna setting, data volume depends on the number of associated users and subcarriers). This transfer time is measured at a highly loaded AP, which runs *iperf* instances to associated users. We observe that even for 16 users with 80 MHz channel (16 to be the max. number of  $V$ s that can be stored in our AP’s hardware cache), coherence time is 44.8% higher than transfer latency. For a less loaded system, over stable channels, coherence time can be up to two orders of magnitude greater than this latency. Our experiments never



**Figure 10:** *MUSE* implementation optimizations. (a) Percentage of channel coherence time spent in PCIe (DMA) transfer. (b) Distribution of coherence bandwidth. (c) CPU execution latency for varying coherence bandwidth and time.

showed any performance degradation due to PCIe transfer overheads.

## 5.2 Complexity Reduction Modules

*MUSE* seeks to further optimize the computational costs coming from (1)  $V$  decomposition, (2) muSINR computation, and (3) user selection. Decompression is required to reverse the Givens Rotation [15] and recover the uncompressed  $V$  matrix. Its overhead depends on the number of antennas on the AP, number of users, and the quantization level. We optimize the decompression operation by eschewing matrix multiplication and sin/cos trigonometric functions, using a lookup-table based approach. With this optimized implementation, the per-subcarrier per-user  $V$  decompression for a 4-antenna AP takes approximately  $10 \mu\text{s}$ . muSINR metric is less computationally expensive since delta SNR is already available for MU-MIMO users. User selection is the least computationally expensive module. Particularly, the distribution of the CPU latency measured in our AP is 84% for decompression, 14% for muSINR calculation and 2% for user selection. Decompression and muSINR computation are performed periodically, per-subcarrier and per-user.

To reduce computational costs, *MUSE* only performs the above operations with the time-granularity approximately equal to coherence time, and frequency-granularity (number of subcarriers) equal to coherence bandwidth. *MUSE* estimates the 50-percentile channel coherence bandwidth and time using uplink CSI measured at the AP, following [8]. Our experiments validate that channel estimation can indeed reduce *MUSE* costs. Fig. 10(b) shows the coherence bandwidth distribution, measured at various locations in the floorplan of Fig. 1. We observe coherence bandwidth to be greater than 4 MHz in 60% of the channels for static users. Coherence bandwidth is smaller for mobile users ( $> 4$  MHz in 10% of the cases). Coherence time is typically greater than 500 ms and can be up to 5s in static settings [26]. Our AP’s CPU execution time drops significantly with higher coherence time and bandwidth, as shown in Fig. 10(c). Specifically, in a scenario where 16, 1-antenna users are connected to our AP, *MUSE* CPU execution time varies from  $182 \mu\text{s}$  to 9 ms, which is much smaller than the typical channel coherence time. We also observe that coherence time has greater impact on CPU execution time compared to coherence bandwidth. Since the AP employ the dual architecture system (Sec. 5), such host CPU execution time does not affect the WLAN packet transmission/reception, which runs on a separate processing unit.

## 6. EVALUATION

In this section, we evaluate *MUSE* performance in a variety of settings, using testbed experiments. Since CSI-based

MU-MIMO user selection and RA designs are not portable in commodity APs (full receiver-CSI is not available), we compare *MUSE* with Legacy-US and PUMA [13]. PUMA selects the best-throughput PHY rate (MCS and spatial streams) and user groups without CSI, by calculating SINR based on the degrees of freedom of each transmission mode as:  $SINR_{puma} = 10 \log_{10} \left( \frac{N_t - N_r + 1}{N_r} \frac{P/N_o}{N_t} \right)$ .  $P/N_o$  is the omnidirectional SNR collected at the AP, and  $N_t$ ,  $N_r$  is the transmit, receive antenna setting. PUMA does not adjust user’s channel bandwidth. We also compare *MUSE* with an optimal user selection algorithm, using trace-driven emulations (Sec. 6.4).

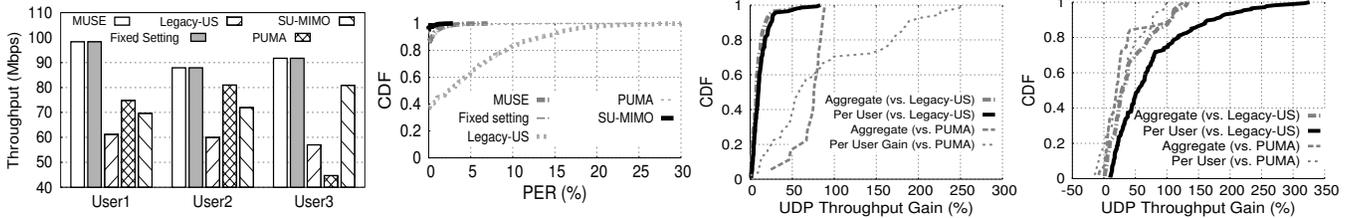
### 6.1 Case Study Comparison

We first evaluate *MUSE* performance in our case-study setting (Sec. 3.3). Our goal is to identify if *MUSE* can overcome MU-MIMO group thrashing problem of Legacy-US, and find the best group and rate setting. In Fig. 11(a) we compare *MUSE* throughput with Legacy-US, PUMA, SU-MIMO (configured for all users) and the best-throughput fixed PHY rate and MIMO (SU- or MU-) mode setting, at 80 MHz. The best-throughput fixed setting is observed when User 1, User 2 form an MU-MIMO group, while User 3 is served in SU-MIMO mode. For this group assignment, the best fixed rates are MCS 5, MCS 7, MCS 9 for User 1, 2 and 3 respectively. Fig. 11(a) shows that *MUSE* gives the same throughput performance as the optimal fixed setting, by selecting the best group and rate, at runtime. *MUSE* does not suffer from inter-user interference and presents almost zero PER, as shown in Fig. 11(b) (each point in the CDF is per-user PER collected in 200ms windows). It gives 46.5% – 61% per-user gains over Legacy-US which suffers from group thrashing, and 13.5% – 41.2% over SU-MIMO mode where channel is time-shared among users.

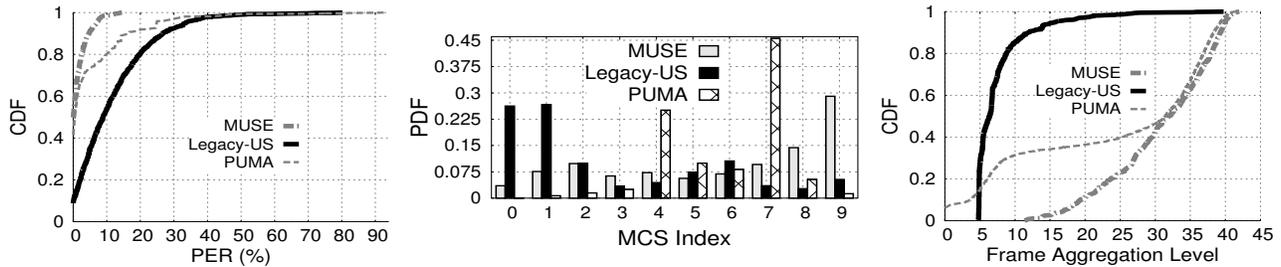
*MUSE* outperforms PUMA with per-user throughput gains from 8.5% to 105.2%. Although PUMA identifies best-throughput MU-MIMO groups, it selects PHY rates lower than the optimal. Specifically, PUMA selects MCS 5 for all the users, while the best-throughput rates for users 2, 3 are MCS 7, MCS 9, respectively. This is because PUMA over-estimates inter-user interference, and computes an  $SINR_{puma}$  value that is lower than the real SINR. Although PUMA’s PER is negligible (cf. Fig. 11(b)), its PHY rate under-selection leads to poor throughput performance.

### 6.2 Controlled Experiments

We further evaluate *MUSE*, Legacy-US and PUMA in multiple controlled, interference-free settings. In all our experiments, 3 smartphone users are placed at multiple spots as shown in Fig. 1 and are connected to AP1. The results presented here are averaged over multiple runs.



**Figure 11:** *MUSE* performance in multiple static and dynamic settings (80 MHz). (a) Throughput performance. (b) PER distribution. (c) UDP gains in static settings. (d) UDP gains in dynamic settings.



**Figure 12:** Analysis of *MUSE* in “high gain” scenario. (a) PER distribution. (b) PHY rate distribution. (c) Frame aggregation.

**UDP traffic:** We first evaluate *MUSE* performance for UDP traffic. Fig. 11(c) shows the throughput gain distribution of *MUSE* w.r.t. Legacy-US and PUMA for static users. *MUSE* always performs similar or better than the other designs, with up to 61.9% and 88.7% aggregated throughput gains over Legacy-US and PUMA, respectively. The median aggregated throughput gains over Legacy-US and PUMA are 7.2% and 76.2%, respectively. Certain users may suffer more from high inter-user interference, and per-user throughput gains can be higher (up to 253% over PUMA).

We further evaluate *MUSE* in more dynamic controlled settings, by moving the 802.11ac smartphones around, and by having people moving in certain patterns. Fig. 11(d) shows the throughput gain distribution of *MUSE* in such settings. Our results show higher throughput gains of *MUSE* over Legacy-US in dynamic, compared to static settings. For example, *MUSE* median and max aggregated throughput gains over Legacy-US are 28.9% and 132.7%, respectively. This is attributed Legacy-US’s limitations to identify the best MU-MIMO group and to converge fast to the best rate. On the other hand, PUMA can converge faster to the best setting since it requires one per-user SINR estimation to select the best group and rate, while Legacy-US uses historical PER statistics. Interestingly, for a small number of cases we observe PUMA to converge faster to the best setting than *MUSE*, and to perform slightly better (cf. Fig. 11(d)).

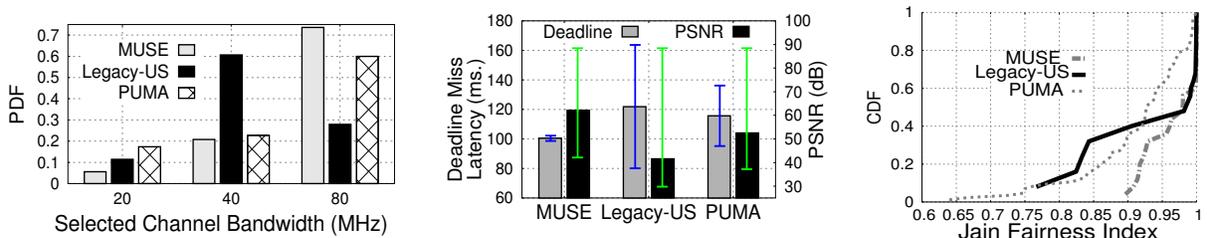
We seek to get more insights about *MUSE* performance, by further analyzing the dynamic scenario where *MUSE* gives the highest aggregate throughput gains over Legacy-US (132.7%) and PUMA (128%). In this scenario, a user is located at spot  $H$  (cf. Fig. 1), while 2 users are moving in a trajectory around  $H$ . *MUSE* achieves these gains by selecting inter-user interference-free MU-MIMO groups, and avoid transmitting at rates above the best-throughput ones. This is shown in PER distribution of Fig. 12(a), where each point in the CDF is the per-user PER collected in 200ms windows. *MUSE* median PER is zero and only in 2% of the samples its PER exceeds 10%. Legacy-US cannot adapt to dynamic environment changes and gives a median and max. PER of 9% and 80.2%, respectively. High PER in Legacy-

US affects its ability to identify the best rate, as shown in the aggregated (from all users) rate distribution of Fig. 12(b). We observe Legacy-US to transmit more than 52% of the frames at the two lowest rate options, while *MUSE* transmits approximately only 10% at these rates. Finally, PUMA inter-user interference estimation is a function of the transmit/receive antennas and never changes with the environment. Consequently, it may underestimate inter-user interference, which results in up to 93% PER (Fig. 12(a)). PUMA’s lack of adaptivity to the changing environment is shown in Fig. 12(b), where PUMA users transmit 70% of their frames at only two rate options, despite the highly dynamic environment.

Interestingly, our results uncover two “side” benefits of *MUSE* over Legacy-US and PUMA, related to *MPDU* frame aggregation, and channel bandwidth adaptation, which highly affect 802.11ac network’s performance.

**Frame aggregation:** The frame aggregation is a key mechanism to amortize protocol overheads, by sending more data in a single transmission. Our results show that *MUSE* can achieve much higher aggregation levels (*i.e.*, number of MPDUs inside an A-MPDU) than Legacy-US and PUMA. Fig. 12(c) shows the aggregation level distribution of the different designs, in the dynamic setting described above (each point is the per-user average aggregation level in 200ms windows). We observe *MUSE* and Legacy-US median aggregation levels to be 32 MPDUs and 6 MPDUs, respectively. *MUSE* slightly outperforms PUMA as well. *MUSE* higher aggression level is attributed to its lower PER compared to the other designs. 802.11 aggregation algorithm uses a TCP-like sliding window to aggregate MPDUs. This window moves forward as long as the MPDUs with sequence numbers inside it have been acknowledged. High PER can affect this window from moving forward, as it has been shown in [19].

**Bandwidth adaptation:** The channel bandwidth adaptation algorithm implemented in our AP, immediately switches to lower bandwidths upon high PER, since it attributes these losses to external interferences (*e.g.*, hidden terminals). It will switch to higher bandwidths when PER becomes low. Inter-user interference from erroneous MU-MIMO group and



**Figure 13:** (a) PDF of selected channel bandwidths. (b) Deadline miss and PSNR of video frames (error bars represent maximum and minimum values). (c) Fairness comparison.

Algorithm Comparison	Gain (%) Distribution	Static		Dynamic	
		Aggregate	Per User	Aggregate	Per User
vs. Legacy-US	Max.	32.2	60.6	80.3	202.8
	Median	12.6	17.5	47.0	96.2
	Min.	0.3	10.5	25.7	39.5
vs. PUMA	Max.	82.3	95.3	70.2	120.4
	Median	20.1	25.2	29.5	40.5
	Min.	2.4	8.4	-4.2	-10.5

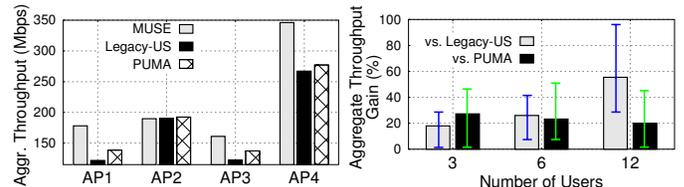
**Table 1:** TCP throughput gains in various settings.

PHY rate selection can increase PER, which will be falsely attributed to external interference factors and will drop channel bandwidth. We illustrate this problem in Fig. 13(a), which shows channel bandwidth distribution of the evaluated designs, in dynamic scenario described above. We observe *MUSE* to transmit the vast majority of the frames (74%) at 80 MHz, compared to other designs which show a lower bandwidth distribution. We further evaluate *MUSE* bandwidth adaptation in Section 6.4.

**TCP traffic:** We next compare the different designs for downlink single-stream TCP flows, in the same static and dynamic settings, as in UDP case. Table 1 summarizes *MUSE* throughput gains. Our results show higher median throughput gains of *MUSE* over Legacy-US for both aggregated and per-user cases. We attribute these higher gains to the negative impact of Legacy-US high PER to TCP congestion control. Similarly, *MUSE* median aggregated, and per-user throughput gains over PUMA are approximately 9% higher than the UDP case, in dynamic settings. However, in static scenarios, *MUSE* gains over PUMA drop due to stable channel.

**Video:** We evaluate the designs' performance for video traffic. In our scenario, a user roaming near the AP, streams a HD (1080p30) quality video, while two users are static and receive UDP traffic. We often observe video stalls for Legacy-US and PUMA due to video frame losses, particularly when group thrashing happens. Dropped video frames can miss their decoding (and display) deadlines. In addition, bit-rate adaptation affects the quality of the received video frames, thus affecting their PSNR. In Fig. 13(b) we show the average (and max./min.) frame deadline miss, and the video PSNR for the different algorithms. Legacy-US and PUMA show up to 65% and 35% higher deadline miss over *MUSE*. Further, *MUSE* achieves on average 9.6 and 20.5 dB higher PSNR than PUMA and Legacy-US respectively.

**Fairness:** We further compare the different designs in terms of throughput fairness. In Fig. 13(c), we present the Jain Fairness Index distribution (in terms of throughput), for all the settings presented above. Each point in the CDF is the fairness index at one setting. Index value of 1 implies perfect fairness. While *MUSE* fairness never drops below 0.9, in 40% and 35% of the settings, Legacy-US and PUMA have a Jain index lower than 0.9, respectively. The lower fairness of Legacy-US and PUMA is attributed to their erroneous MU-



**Figure 14:** Performance in field-trials. (a) Aggregate throughput per AP. (b) Gain scaling (error bars represent maximum and minimum values).

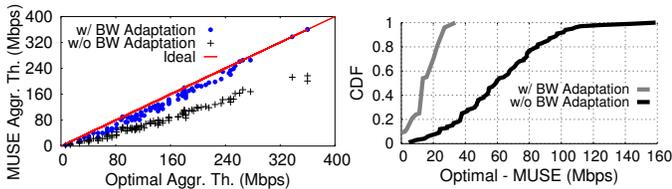
MIMO grouping, where inter-user interference often affects only a subset of the three users of a group. These users typically get lower throughput than the others. *MUSE* does not always get perfect fairness, since our AP has been designed to provide air-time (and not throughput) fairness. So, users located far away from the AP get the same air-time, but achieve lower throughput than the ones closer to the AP.

### 6.3 Field Trial

We next evaluate *MUSE* in realistic field trials, where various sources of dynamics coexist in a complex manner. We setup 4 APs (AP1 - AP4), (see Fig. 1 - AP4 is outside the floorpan), and we connect 15 smartphones to them. We generate downlink traffic from the APs to the users; 10 users receive UDP, 4 users receive single-stream TCP, and 1 user downloads video traffic. All the users are static for AP3 and AP4, while for each of the AP1, AP2, two users are moving at pedestrian speed. We perform our experiments during working hours, where many people are walking around. Apart from our APs, we detect 58 other APs (radios) during our experiments, where 26 of them operate on the 5 GHz band. These APs switch channel periodically and often operate on channel 149, which is used by our 802.11ac APs.

We perform 15-minute experiments for multiple runs, and we compare *MUSE* performance with Legacy-US and PUMA. Fig. 14(a) shows the designs' aggregate throughput for each AP. We observe *MUSE* throughput gains to be 29% - 46.5% and 17.2% - 28.6% over Legacy-US and PUMA respectively, for AP1, AP3 and AP4. *MUSE* throughput gains can go up to 54.8% and 61.4% over Legacy-US and PUMA for individual users. All the designs perform the same at AP2. This is attributed to the high multi-path richness of the area around AP2 (many objects located in this area), that makes all the designs to operate at MU-MIMO with appropriate grouping, which overall performs well.

Further, we vary the number of users connected to a single AP to understand the gain scaling of *MUSE* w.r.t. Legacy-US and PUMA. We place the AP to 8 arbitrary locations in the floorplan (Fig. 1) and for each locations, we connect up to 12 static and mobile users. Fig. 14(b) shows the aggregate throughput gain (%) of *MUSE* w.r.t. Legacy-US and



**Figure 15:** *MUSE comparison with optimal user selection. (a) Throughput comparison in multiple topologies. (b) Distribution of throughput gap between MUSE and optimal.*

PUMA. *MUSE* achieves up to 96.2% aggregate gain compared to Legacy-US. While the aggregate gain is only up to 51% *w.r.t.* PUMA, per-user gain can go up to 197.8% within the experimental setting.

## 6.4 Comparison with Optimal User Selection

We finally compare *MUSE* with an optimal user selection algorithm, which leverages *full CSI* and applies an exhaustive search to identify the best groups. We evaluate *MUSE* with its bandwidth adaptation module (cf. Sec. 4.3) turned on and off<sup>5</sup>. Since the optimal algorithm requires full CSI which is not supported by 802.11ac APs, we use trace-driven emulation instead. We first collect CSIs from users located in multiple spots (Fig. 1). We then emulate multiple topologies, each containing 10 randomly located users, connected to one AP. For a given topology, we emulate bandwidth heterogeneity by randomly selecting a bandwidth option for each user. The scatter plot 15(a) compares optimal and *MUSE* aggregated throughputs, where each point represents one topology. As we further show in Fig. 15(b), the throughput gap between optimal and *MUSE* is less than 25 Mbps in 80% of the topologies, when the bandwidth adaptation is on. This corresponds to at most 21% throughput loss in 80% of the topologies and a median loss of only 12%. Note that our emulation does not count the overheads of feeding *full CSI* back to the AP. Hence, we expect this throughput loss to be much smaller in reality. Fig. 15(b) further shows the performance gains from *MUSE* bandwidth adaptation. Upon disabling the bandwidth adaptation module, median throughput gap between optimal and *MUSE* grows from 15 Mbps to approximately 60 Mbps. The median throughput loss after disabling *MUSE* bandwidth module is 59%, in contrast to 14% in adaptation enabled case.

## 7. RELATED WORK

**MU-MIMO user selection:** The theoretical capacity of MU-MIMO systems has been widely studied [5, 27], and can be realized when full receiver’s CSI is available at the AP. There are multiple algorithms that leverage CSI to perform MU-MIMO user selection [3–7, 25]. However, communicating full receiver’s CSI to the AP incurs large overheads, which grow with the number of OFDM subcarriers and the number of transmitter-receiver antenna pairs [8]. To reduce such overhead, one approach is the quantization (compression) of the CSI feedback [5, 7, 8]. For example, AFC [8] quantizes CSI 3 dimensions: time, frequency and numerical values. Alternatively, the AP can collect of CSI less frequently, from limited number of users [9, 25, 28]. Finally, recent proposals [10] seek to reduce various protocol overheads, such as long 802.11 MU-MIMO contention periods.

<sup>5</sup>When bandwidth module is off, *MUSE* buckets users of same bandwidths, and runs greedy search on each bucket.

All the above approaches require users’ full CSI feedback and are not directly applicable to 802.11ac systems. However, *MUSE* can still leverage these algorithms to further reduce sounding overheads.

Recent proposals seek to perform user selection without CSI. PUMA [13] selects the best MU-MIMO group, using a theoretical MU-MIMO capacity scaling model. Our experiments showed that PUMA cannot properly capture inter-user interference in practical settings. Argos [29] applies implicit beamforming, where uplink pilots are used for downlink beamforming. Different from Argos, 802.11ac supports only explicit beamforming. Finally, Signpost [30] evades CSI by broadcasting predefined orthogonal vectors. Signpost applies only to uplink MU-MIMO, which is not supported by 802.11ac.

**Rate adaptation in MU-MIMO:** Rate adaptation has not been widely explored in 802.11ac. The most relevant work to ours is TurboRate [31], a cross-layer RA design for uplink MU-MIMO that requires customized MAC/PHY functionalities. To our knowledge, our study is the first to show the joint effect of user selection and RA in 802.11ac systems.

**Distributed MU-MIMO:** The distributed MU-MIMO allows multiple APs operating on the same channel, to transmit concurrently to multiple receivers, without interfering with each other [32–34]. Such solutions require synchronization among APs. Different from these efforts, *MUSE* is a fully distributed design, which operates at a single AP. MIDAS [35] is a distributed antenna system which seeks to leverage the full potential of 802.11ac MU-MIMO networks. Different from *MUSE*, MIDAS selects MU-MIMO users based on their antenna preference (based on the average signal strength) and fairness. Finally, systems such as Geosphere [11] propose modifications in the PHY layer to minimize errors from interference among MIMO streams. *MUSE* is fully implemented in MAC, and does not require any modifications in PHY layer.

## 8. CONCLUSION

In this paper, we use commodity 802.11ac APs to study MU-MIMO user selection protocols. Our results show that the limited feedback provided by 802.11ac users, the heterogeneous bandwidth users and the limited-resource APs, pose significant challenges in designing and implementing such protocols. To this end, we propose *MUSE*, a new user selection framework, which leverages the limited compressed beamforming feedback from 802.11ac users, to identify the best-throughput MU-MIMO groups. *MUSE* can increase the MU-MIMO grouping opportunities by jointly adjusting the channel bandwidth with user selection. *MUSE* is lightweight and portable to resource-constraint APs, by leveraging a new firmware-kernel interaction architecture. Our work is the first to optimize MU-MIMO performance in 802.11ac commodity devices, and we consider it as an important milestone to design the future 802.11ax and 5G wireless networks.

## Acknowledgement

We sincerely thank the anonymous reviewers and our shepherd for their valuable comments and feedback. Sanjib Sur and Xinyu Zhang were partially supported by the NSF under Grant CNS-1506657, CNS-1518728, CNS-1343363 and CNS-1350039 during this work.

## 9. REFERENCES

- [1] M. X. Gong, B. Hart, and S. Mao, "Advanced Wireless LAN Technologies: IEEE 802.11ac and Beyond," in *ACM GetMobile: Mobile Comp. and Comm.*, 2015.
- [2] "Ericsson 5G field trial gear achieves peak downlink throughput over 25 Gbps with MU-MIMO," in *Ericsson press release*, 2016.
- [3] M. Esslaoui, F. Riera-Palou, and G. Femenias, "A fair MU-MIMO scheme for IEEE 802.11ac," in *ISWCS*, 2012, pp. 1049–1053.
- [4] Z. Shen, R. Chen, J. G. Andrews, R. W. Heath, and B. L. Evans, "Low Complexity User Selection Algorithms For Multiuser MIMO Systems with Block Diagonalization," *IEEE Transaction on Signal Processing*, vol. 54, no. 9, 2006.
- [5] T. Yoo and N. J. A. Goldsmith, "Multi-Antenna Downlink Channels with Limited Feedback and User Selection," *IEEE JSAC*, vol. 25, no. 7, 2007.
- [6] T. Ji, C. Zhou, S. Zhou, and Y. Yao, "Low Complex User Selection Strategies for Multi-User MIMO Downlink Scenario," in *Proc. of IEEE WCNC*, 2007.
- [7] D. Gesbert, M. Kountouris, R. W. Heath, and C.-B. Chae, "Shifting the MIMO Paradigm," *IEEE Signal Processing Magazine*, vol. 24, no. 5, 2007.
- [8] X. Xie, X. Zhang, and K. Sundaresan, "Adaptive Feedback Compression for MIMO Networks," in *Proc. of ACM MobiCom*, 2013.
- [9] X. Xie and X. Zhang, "Scalable User Selection for MU-MIMO Networks," in *Proc. of IEEE INFOCOM*, 2014.
- [10] T.-W. Kuo, K.-C. Lee, K. C.-J. Lin, and M.-J. Tsai, "Leader-Contention-Based User Matching for 802.11 Multiuser MIMO Networks," in *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, 2014.
- [11] K. Nikitopoulos, J. Zhou, B. Congdon, and K. Jamieson, "Geosphere: Consistently Turning MIMO Capacity into Throughput," in *ACM SIGCOMM*, 2014.
- [12] "AR9331 Highly-Integrated and Cost Effective IEEE 802.11n 1x1 2.4 GHz SoC for AP and Router Platforms," in *Atheros Data Sheet*, 2010.
- [13] A. Narendra, J. Lee, S.-J. Lee, and E. W. Knightly, "Mode and User Selection for Multi-User MIMO WLANs without CSI," in *IEEE INFOCOM*, 2015.
- [14] IEEE Standards Association, "IEEE Standards 802.11ac-2013: Enhancements for Very High Throughput for Operation in Bands below 6 GHz," 2013.
- [15] H. Lou, M. Ghosh, P. Xia, and R. Olesen, "A Comparison of Implicit and Explicit Channel Feedback Methods for MU-MIMO WLAN Systems," in *IEEE PIMRC*, 2013.
- [16] P. Wang and L. Ping, "On Maximum Eigenmode Beamforming and Multi-User Gain," in *IEEE Transactions On Information Theory*, vol. 57, no. 7, 2011.
- [17] Xiaomi Technology Co. Ltd., "Xiaomi Mi 4i," 2015.
- [18] Atheros, in *Minstrel Rate Adaptation*, 2009.
- [19] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "MIMO Rate Adaptation in 802.11n Wireless Networks," in *ACM MobiCom*, 2010.
- [20] S. H. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust Rate Adaptation for 802.11 Wireless Networks," in *ACM MobiCom*, 2006.
- [21] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
- [22] J. A. Gubner, "Probability and Random Processes for Electrical and Computer Engineers." Cambridge University Press, 2006.
- [23] G. Judd, X. Wang, and P. Steenkiste, "Efficient Channel-aware Rate Adaptation in Dynamic Environments," in *ACM MobiSys*, 2010.
- [24] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," in *ACM SIGCOMM*, 2010.
- [25] W.-L. Shen, K. C.-J. Lin, M.-S. Chen, and T. Kun, "SIEVE: Scalable User Grouping for Large MU-MIMO Systems," in *IEEE INFOCOM*, 2015.
- [26] S. Sen, B. Radunovic, J. Lee, and K.-H. Kim, "CSpy: Finding the Best Quality Channel without Probing," in *ACM MobiCom'13*, 2013.
- [27] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity Limits of MIMO Channels," *IEEE JSAC*, vol. 21, no. 5, 2003.
- [28] O. Bejarano, E. Magistretti, O. Gurewitz, and E. W. Knightly, "MUTE: Sounding inhibition for MU-MIMO WLANs," in *IEEE SECON*, 2014.
- [29] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: Practical Many-Antenna Base Stations," in *Proc. of ACM MobiCom*, 2012.
- [30] A. Zhou, T. Wei, X. Zhang, M. Liu, and Z. Li, "Signpost: Scalable MU-MIMO Signaling with Zero CSI Feedback," in *Proc. of ACM MobiHoc*, 2015.
- [31] 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.
- [32] H. Rahul, S. Kumar, and D. Katabi, "MegaMIMO: Scaling Wireless Capacity with User Demands," in *ACM SIGCOMM*, Helsinki, Finland, August 2012.
- [33] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "AirSync: Enabling Distributed Multiuser MIMO With Full Spatial Multiplexing," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, 2013.
- [34] 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.
- [35] J. Xiong, K. Sundaresan, K. Jamieson, M. Khojastepour, and S. Rangarajan, "MIDAS: Empowering 802.11ac Networks with Multiple-Input Distributed Antenna Systems," in *ACM CoNEXT*, 2014.