Adaptive Feedback Compression for MIMO Networks

Xiufeng Xie and Xinyu Zhang
University of Wisconsin-Madison
{xiufeng,xyzhang}@ece.wisc.edu

Karthikeyan Sundaresan
NEC Laboratories America
karthiks@nec-labs.com

ABSTRACT

MIMO beamforming technology can scale wireless data rate proportionally with the number of antennas. However, the overhead induced by receivers' CSI (channel state information) feedback scales at a higher rate. In this paper, we address this fundamental tradeoff with Adaptive Feedback Compression (AFC). AFC quantizes or compresses CSI from 3 dimensions — time, frequency and numerical values, and adapts the intensity of compression according to channel profile. This simple principle faces many practical challenges, *e.g.*, a huge search space for adaption, estimation or prediction of the impact of compression on network throughput, and the coupling of different users in multi-user MIMO networks. AFC meets these challenges using a novel cross-layer adaptation metric, a metric extracted from 802.11 packet preambles, and uses it to guide the selection of compression intensity, so as to balance the tradeoff between overhead reduction and capacity loss (due to compression). We have implemented AFC on a software radio testbed. Our experiments show that AFC can outperform alternative approaches in a variety of radio environments.

Categories and Subject Descriptors

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

General Terms

Algorithms, Design, Theory, Performance

Keywords

Multi-user MIMO networks, limited feedback, feedback compression

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

1. INTRODUCTION

MIMO beamforming, also referred to as *closed-loop MIMO* technology, has the potential to scale wireless data rate proportionally with the number of transmit antennas. In fact, it has become a hallmark of recent high-throughput 802.11 standards. In particular, 802.11n adopted single-user MIMO (SU-MIMO) beamforming, where multiple transmit antennas *precode* one packet, so that the precoded signals coherently combine at a receiver to enhance SNR. The ongoing 802.11ac further incorporates multi-user beamforming (MU-MIMO), which precodes different receivers' packets, so that they can be transmitted simultaneously without interference. The precoding operations in SU- and MU-MIMO are essentially a weighted combination of outgoing packets, and the weights must be carefully designed based on the channel state information (CSI), *i.e.*, the channel matrix from transmit antennas to all receive antennas. Hence, a transmitter must obtain timely CSI feedback from all receivers. As CSI is needed for different transmit antennas, frequency bands and receivers, the resulting information grows with a multiplication of all these parameters, which causes formidable overhead and may even nullify the MIMO capacity gain. Therefore, managing feedback overhead is critical to the performance of MIMO beamforming.

The 802.11n/ac standards propose to compress the CSI along three dimensions: time — sending CSI less frequently; frequency— allowing adjacent spectrum units (called *subcarriers*) to share the same CSI; and numerical values — quantizing CSI values into a small number of bits. Though promising for reducing feedback overhead, such compression comes with a loss in CSI accuracy and may adversely affect the MIMO network throughput. Therefore, it is critical to strike a balance between overhead reduction and capacity loss. But finding such a sweet spot is non-trivial as it involves sophisticated coupling of all three dimensions. Similar to the bit-rate options, the choice of compression options and levels remains vendor-specific in 802.11, and to the best of our knowledge has not been explored before.

This paper attempts to fill the gap in this space through: a comprehensive measurement study of the impact of compression on network performance, and the design of Adaptive Feedback Compression (AFC) - a feedback management mechanism for SU-/MU-MIMO networks. Our measurement study is based on a comprehensive software-radio implementation of the 802.11n SU-MIMO and 802.11ac MU-MIMO OFDM PHY layer. The results reveal that although CSI compression can save substantial overhead, it may significantly degrade link capacity if too aggressive. More im-

portantly, a link's tolerance to CSI compression depends not only on its channel stability, but also on SNR, spectrum location, and user pairing (for MU-MIMO) — a number of factors unreported in previous works. This calls for an automatic configuration scheme that maximizes compression intensity without losing too much capacity.

AFC is designed to meet this goal with a simple principle. It quantizes or compresses CSI from 3 dimensions: time, frequency and numerical values, and adapts the intensity of compression according to channel profile. Realizing this principle involves non-trivial challenges: (1) The 3 dimensions together constitute a huge parameter space. Hence a comprehensive search algorithm is inefficient and may never converge. In fact, when searching over one dimension, the channel profile w.r.t. the other two dimensions may have already changed. (2) Compression intensity can be configured according to a certain metric that reflects channel stability over time or frequency. Coherence time/frequency is widely used for this purpose. But through experimental study, we find that there does not exist a fixed mapping between such a metric and the optimal compression intensity. (3) An efficient adaptation algorithm must be able to estimate the link capacity change due to CSI compression over a certain dimension. However, the effects of different dimensions on capacity loss are coupled.

AFC meets these challenges using *CNo*, a novel cross-layer adaptation metric that can model the information loss (or errors) due to compression. Upon each packet arrival, AFC can instantaneously extract the CNo metric from the CSI of 802.11 packet preambles, and evaluate the errors and SNR reduction caused by each dimension of compression respectively. As CNo decouples the 3 dimensions of adaptation, it substantially reduces the search space. It also helps isolate the effects of compression from channel fading, which can both result in changes in link capacity. Based on CNo, AFC can easily identify the appropriate compression level that optimizes the overall network throughput, taking into account both capacity loss and overhead reduction.

We have implemented AFC on top of the SU-/MU-MIMO OFDM module that we built on WARP. This PHY module closely follows the 802.11n/ac specifications, and realizes linear precoding based beamforming, OFDM modulation/demodulation, CSI estimation as well as symbol-level decoding. Experiments on a WARP testbed show that AFC's adaptation metric leads to fast and effective selection of compression levels along all 3 dimensions. In contrast, no single static configuration of compression intensity can fit all channel/network profiles. Overall, by balancing the capacity-overhead tradeoff, AFC's *throughput performance* is comparable with the best fixed compression scheme for any given channel profiles, and 40% to 120% higher than the non-adaptive scheme in 802.11n/ac.

In summary, we make the following contributions in this paper.

- We present the first comprehensive measurement study of the effects of feedback compression in 802.11 SU-/MU-MIMO networks.
- We propose AFC, which introduces effective cross-layer metrics to trigger the adaptation of compression levels along 3 dimensions, and strike a balance between overhead reduction and capacity loss.

To our knowledge, AFC represents the first algorithm that provides guidelines to address this tradeoff for SU-/MU-MIMO beamforming in wireless networks.

2. BACKGROUND

2.1 SU-/MU-MIMO networks

MU-MIMO systems are commonly realized use Zero-Forcing Beamforming (ZFBF), a low-complexity linear precoding scheme that is amenable for practical transceivers. Consider a network with a n-antenna AP and n single-antenna receivers. Let $\boldsymbol{X}$ be a $n \times 1$ vector, representing the data symbols to be sent to n receivers, then the vector of received symbols corresponding to the n receivers is:

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{W}\boldsymbol{X} + \boldsymbol{V} \tag{1}$$

where $\boldsymbol{H}$ is the $n \times n$ channel matrix, and $\boldsymbol{V}$ the $n \times 1$ noise vector. $\boldsymbol{W}$ is the precoding matrix, with each row W_{*k} corresponding to the precoding vector of transmit antenna k. In ZFBF, $\boldsymbol{W}$ is obtained via a pseudo-inverse of the channel matrix:

$$\boldsymbol{W} = \boldsymbol{H}^* \cdot (\boldsymbol{H} \cdot \boldsymbol{H}^*)^{-1} \tag{2}$$

where $(\cdot)^*$ is the complex conjugate operator.

With precoding, it is straightforward that the received symbols become $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{V}$, *i.e.*, data symbols of different receivers are decoupled — each receiver only obtains its own symbol plus some noise. In practical MIMO transceivers, each antenna is accompanied by one radio back-end and has a maximum power constraint P_m. To satisfy this constraint, the precoding vector of each transmit antenna k should be scaled by: $\arg\max_j \sqrt{P_m}/||W_{*j}||$ [1].

SU-MIMO can be considered as a MU-MIMO network with one receiver but n transmit antennas. The pseudo-inverse precoding will result in n copies of the receiver's symbols added up, boosting the signal power to n^2.

To perform precoding, however, both SU- and MU-MIMO transmitters need to know the CSI, *i.e.*, the channel matrix $\boldsymbol{H}$, which can only be obtained from receivers' feedback. In this paper, we only consider precoding for single-antenna receivers, but multi-antenna precoding is also feasible through block-diagonalization [2]. The adaptation algorithm proposed in this paper is not limited to any precoding scheme as it only leverages vanilla 802.11 preambles.

2.2 CSI feedback and compression in 802.11 SU-/MU-MIMO

In 802.11n/ac, before beamforming a data packet, a SU-/MU-MIMO transmitter follows a polling procedure to obtain CSI from the receiver(s). Fig. 1 illustrates an example for a MU-MIMO network with 2 transmit antennas and 2 receivers. The transmitter first sends an NDP announcement to initiate the procedure, and then polls the intended receivers one by one and obtains CSI feedback from them. Afterwards it precodes the data packets based on the CSI. CSI feedback for SU-MIMO follows the same procedure.

The accuracy of CSI determines how effectively the inter data-symbol interference can be cancelled for MU-MIMO, and whether copies of a symbol can align their phases to enhance received SNR in SU-MIMO. Hence, it is critical to the SNR and consequently network capacity.

However, accuracy comes at the cost of overhead. To bound the CSI overhead, 802.11 allows CSI to be compressed,

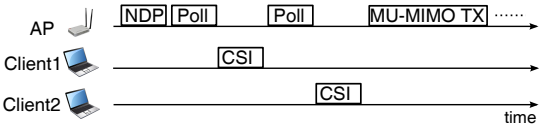

Figure 1: CSI feedback and MU-MIMO beamforming in 802.11ac.

with multiple levels of intensity, along three dimensions: i) Time, *i.e.*, instead of per-packet feedback, allowing CSI to be shared among multiple subsequent packets. ii) Frequency, *i.e.*, instead of collecting CSI for each subcarrier (the smallest spectrum unit in an 802.11 OFDM band), group neighboring subcarriers and use one subcarrier's CSI to represent its neighbors'. iii) Numerical values, *i.e.*, instead of sending float-point complex numbers directly, quantizing the CSI into a small number of bits.

To compress CSI in frequency-domain, 802.11 allows 1, 2, or 4 subcarriers to be grouped and share the same CSI[1]. In addition, for each CSI value, its real and imaginary components can be quantized into 4, 5, 6 or 8 bits [3]. As channel stability and inter-packet time are hard to characterize, 802.11n/ac employs per-packet CSI feedback, leaving the time-domain compression as an open option.

3. MOTIVATION

In this section, we use simulation and testbed measurement to evaluate the MAC-layer overhead reduction and PHY-layer capacity loss due to compression, which motivates the need for an mechanism that automatically configures compression intensity to address this tradeoff.

3.1 How much can compression save?

We provide an account of all factors that contribute to the CSI feedback *overhead* – defined to be the ratio between the time spent in obtaining CSI (including the NDP and polling packets) and that in data transmission. To this end, we extend the ns-2 simulator with a SU-/MU-MIMO MAC module, following the 802.11n/ac MAC specifications. We then evaluate the CSI overhead for UDP transmissions in a MIMO WLAN. By default, the simulation is configured to 20MHz bandwidth (with 52 data subcarriers), 1.5KB packet size, 4 transmit antennas and 1 receiver, 32-bit float-point CSI (for both real and imaginary components), and per-packet CSI feedback. We first isolate the PHY layer effect by configuring an intermediate data rate of 18Mbps for all receivers. Note that CSI feedback is always sent with 6Mbps data rate in 802.11. Fig. 2 plots the CSI overhead when varying each factor while keeping others to the default values.

Bandwidth (number of subcarriers). CSI overhead increases linearly with the spectrum width, which in turn depends on the number of subcarriers. For a typical 20MHz spectrum, the CSI feedback overhead can be 3.4× compared to data transmission, and increases linearly with bandwidth (Fig. 2(a)). Note that the NDP, polling packets, and their inter-frame spacing also contribute to the overhead. However, these are fixed, resulting in the CSI overhead dominating. By compressing CSI in frequency domain, the feedback overhead decreases sharply, on the same order as the number of subcarriers grouped together to share CSI.

[1]A 20MHz band in 802.11 contains 52 data subcarriers. The number of subcarriers increases linearly with bandwidth.

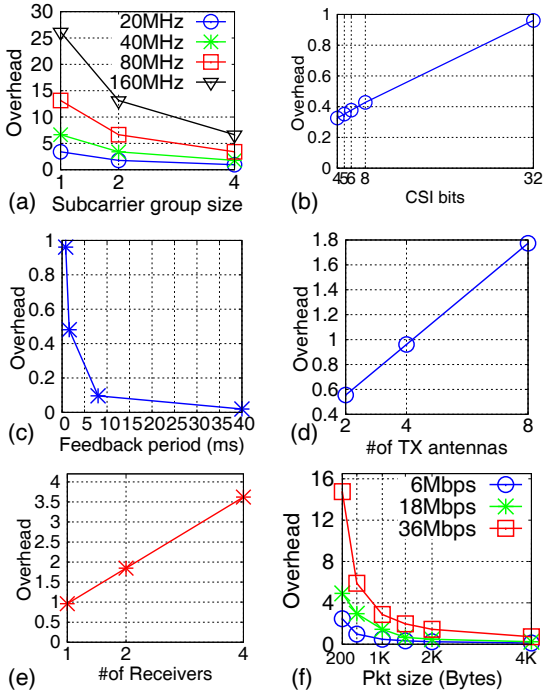

Figure 2: Various parameters affect CSI feedback overhead. y-axis is the ratio between channel time usage of CSI feedback and that of data frames.

CSI bits. Representing CSI with full-resolution float-point numbers is clearly inefficient (Fig. 2(b)) — it induces an overhead of 0.96 under the default network setting. Quantization of the numerical values can reduce the overhead almost linearly, to 0.33 with the most aggressive 4-bit quantization defined in 802.11.

Feedback period. Similar to the above two factors, by compressing the CSI over time, *i.e.*, increasing the feedback period, an inverse-linear reduction in overhead is achieved, as shown in Fig. 2(c). CSI feedback overhead is only 0.096 with an 8ms feedback period and 0.019 with 40ms. Note that our simulation isolates the compression along these three dimensions. As they can be done independently, a combination would achieve a multiplicative reduction.

Number of TX antennas and receivers. These two factors independently account for an almost linear increase of overhead (Fig. 2(d) and (e)), as the channel from each transmit antenna to each receiver needs to be estimated and fed back. Unfortunately, no CSI compression is feasible w.r.t. these two factors.

Data rate. For small packet sizes, CSI induces formidable overhead, *e.g.*, 2.46× for a 200B and 0.48 for a 1KB packet sent at 6Mbps (Fig. 2(f)). Increase of data rate amplifies the relative overhead, to 2.88× for a 1KB packet at 36Mbps.

From the above simulation results, we can infer that the relative overhead of CSI feedback is roughly linear w.r.t.:

$$\frac{\#\text{SubcGroups}\times\#\text{CSIbits}\times\#\text{TXantennas}\times\#\text{Receivers}\times\text{DataRate}}{\text{Feedback period}\times\text{Pkt size}}$$

Among these factors, the number of subcarrier groups, CSI bits and feedback period can be compressed. 802.11's standard linear compression over the three dimensions can contribute to a multiplicative reduction in feedback overhead. With more sophisticated compression schemes, such as a *Givens rotation* [4, Section 20.3.12] of channel matrix,

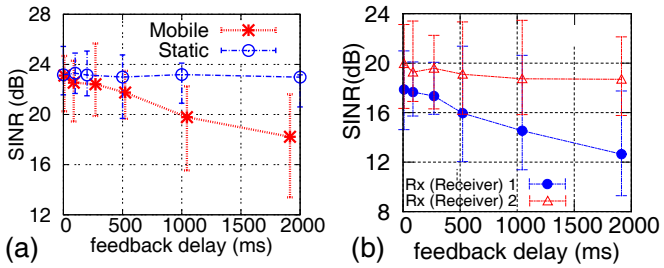

Figure 3: Effects of time-domain compression on SNR for (a) SU-MIMO and (b) MU-MIMO. Error bars show the max and min SNR among 10 experimental runs, unless noted otherwise.

feedback overhead can be further reduced. Besides compression, frame aggregation can also reduce the *relative overhead* of CSI by amortizing it over concatenated data frames. But large frames may incur unacceptably long buffering delay, which compromises the performance of real-time applications such as VoIP and interactive web sessions. Such effects limit the practical usage of frame aggregation.

We will now study the associated impact of feedback compression on the link capacity of MIMO networks.

3.2 Why adaptive compression?

Feedback compression, if too aggressive, can severely degrade the accuracy of CSI and lower the achievable SNR or bit-rate. The extent to which compression affects network performance varies with the radio environment (channel profile), which we will study through comprehensive experiments below. Our experiments are conducted on a testbed of 5 WARP MIMO software radios, located in an office building. We have implemented a full-fledged MIMO-OFDM communication module on WARP, following the 802.11n/ac specification for SU- and MU-MIMO networks (Sec. 5 provides a detailed description of our implementation). With this testbed implementation, we perform a quantitative study to identify the channel characteristics that lead to the positive and negative effects of CSI compression. These in turn contribute to the factors that AFC should monitor and react appropraitely.

Effects of time domain compression. The time-domain accuracy of compressed CSI depends on how frequently it is refreshed, relative to the channel variation over time. In Fig. 3, we allow a transmitter to periodically initiate CSI polling, obtain feedback from receiver(s), and send one beamforming packet. We then compute the SNR[2] by varying the feedback period.

We consider two scenarios: i) static environment: a research lab with negligible human movement. ii) mobile nodes: receiver(s) are moving around the transmitter (along a circle) at walking speed. From the results in Fig. 3(a), we see that in static environment, SU-MIMO link SNR is virtually unaffected by time-domain compression, even with a 2-second feedback period. This implies substantial space for CSI overhead saving: for each link, packets separated by within 2 seconds can share the same CSI.

[2]Our testbed implementation allows us to obtain SNR from decoded symbols. This leads to an accurate estimation, which accounts for not only ambient noise, but also receiver-induced noise due to, *e.g.*, imperfect channel estimation.

The impact of stale CSI becomes pronounced as the radio environment becomes more dynamic, as reflected in the large variation of SINR. When nodes are moving at walking speed, we observed that average SINR may reduce by 1 to 2 dB when feedback period is below 300ms. But the reduction may vary significantly across different experiments. We have observed in certain experiments that even a 20ms feedback period can reduce SNR by up to 5.3dB compared with an oracle scheme with zero feedback delay.[3] Clearly, an aggressive time-domain compression will hurt the MIMO beamforming performance for mobile nodes. However, due to packet transmission delay (*e.g.*, caused by contention), inter-packet arrival time can easily reach 20ms. Hence, *per-packet feedback becomes inevitable for mobile nodes, which necessitates compression from alternative dimensions.*

Fig. 3(b) shows the performance of a MU-MIMO network with a 2-antenna AP and two receivers (one mobile, the other static). MU-MIMO is more sensitive to CSI than SU-MIMO, since the channel variation of multiple receivers are independent, and therefore their CSI errors add up and exacerbate the impact of stale CSI. Remarkably, mobility only affects the mobile receiver itself. This is because when running linear precoding, the transmitter only needs to ensure signals from all its antennas combine coherently at each intended receiver, which is only related with the channel gains between transmit antennas and that receiver. This also implies that *in MU-MIMO networks, different receivers can perform time-domain compression and adaptation independent of each other, which is critical for a scalable solution.*

Effects of frequency domain compression. Theoretical wireless channel models [5] have shown that neighboring subcarriers of a wide spectrum may be highly correlated with each other, hence it is sufficient to use one subcarrier's CSI to represent that of a group. Subcarrier correlation is roughly inversely proportional to multi-path delay spread, *i.e.*, the maximum separation between multiple reflected copies of a signal. In environment with rich reflections and obstacles, delay spread is small and channel may be correlated over a wide spectrum with many subcarriers, making it possible to compress CSI aggressively in frequency domain. Our experiments in Fig. 4 indeed verify this effect. In a small laboratory environment, compressing neighboring 4 subcarriers results in negligible SNR loss. A lobby – a relatively open-space environment, experiences a larger delay spread (consistent with theoretical channel models [5]) and is hence more sensitive to compression. Compressing 4 subcarriers degrades SNR by 3 dB on average and up to 3.3dB in certain cases (MU-MIMO in Fig. 4 (b)).

Interestingly, our experiments reveal several other factors governing the effectiveness of frequency-domain compression. First, different spectrum bands (with different carrier frequencies) reveal different levels of sensitivity to subcarrier grouping. A typical 2.4GHz band exhibits high sensitivity, since it may experience different level of leakage noise (interference) from neighbors across its spectrum. Even across the 5GHz bands, which are unused around our testbed, the sensitivity varies. We suspect this to be the result of imperfect radio hardware, which causes unequal RF/antenna gains across the spectrum. Finally, mobility does not affect the impact of frequency-domain compression, implying it can

[3]We create the oracle case by collecting CSI and then replaying the channel distortion effect (Sec. 5). The transmitter is an oracle which knows the CSI without polling and feedback.

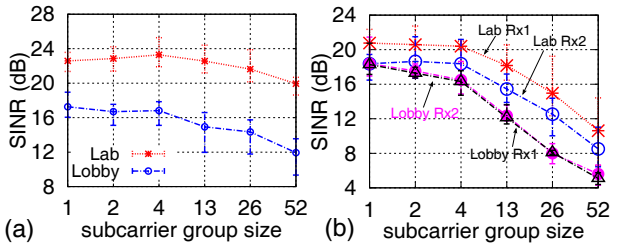

Figure 4: Effects of subcarrier grouping on the performance of (a) SU-MIMO and (b) MU-MIMO.

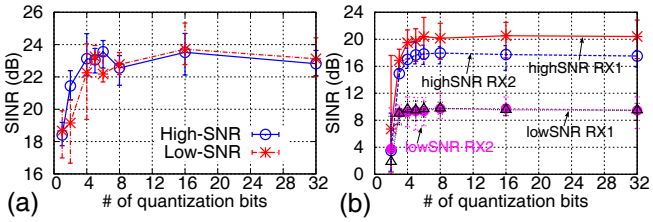

Figure 5: Effects of CSI quantization on the performance of (a) SU-MIMO and (b) MU-MIMO.

be decoupled from time-domain compression. MU-MIMO is more sensitive to subcarrier grouping than SU-MIMO, for the same reason as in time-domain compression.

Effects of numerical quantization. Fig. 5 plots the MIMO network capacity as a function of the number of quantization bits used to represent CSI[4]. It reveals a sub-linear relation: capacity grows sharply as quantization bits increases from 2 to 3, but shows marginal improvement beyond that. In addition, low SNR links are less sensitive to quantization, as channel noise dominates the quantization errors. Again, MU-MIMO is more sensitive to quantization due to the additive quantization noise of multiple receivers.

To summarize, although compression can save substantial overhead, it may come at the cost of capacity loss. This tradeoff manifests differently depending on the radio environment and network dynamics. Clearly, an autonomous algorithm is needed to guide the 3-D compression and hence balance the overhead reduction and capacity loss, thereby resulting in maximium network throughput.

4. ADAPTIVE FEEDBACK COMPRESSION

4.1 Overview

AFC is a systematic approach to balancing CSI compression intensity with SU-/MU-MIMO network capacity, thereby helping achieve optimized network throughput under various network (channel) conditions. The core component of AFC is a cross-layer adaptation algorithm that extracts the channel dynamics from 802.11 packet preambles, and approximates a throughput-optimal point by choosing an appropriate compression configuration over 3 dimensions: time, frequency and numerical values.

Cross-layer adaptation. Fig. 6 showcases AFC's adaptation procedure for a SU-MIMO network. CSI feedback is reactive in AFC. By default, the transmitter uses the previous (most recent) CSI feedback for beamforming precoding.

[4]802.11 defines three algorithms for quantizing CSI values. Without loss of generality, we adopt the direct quantization approach [4, Section 20.3.12.2.1] which compresses the real and imaginary components into a few number (4, 5, 6 or 8) of bits.

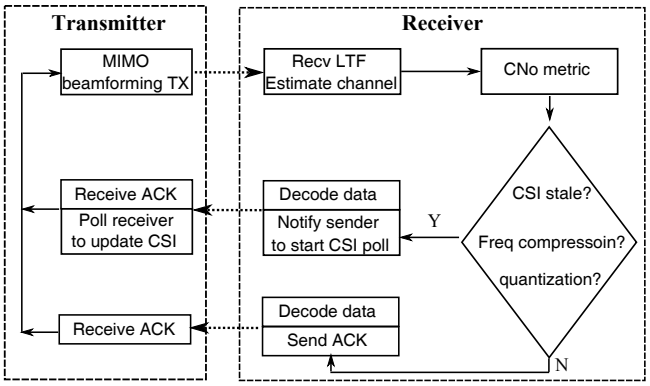

Figure 6: Cross-layer adaptation protocol in AFC.

It initiates the CSI polling only if the receiver indicates a need to do so. The receiver is able to refresh its channel estimation whenever it receives a new packet. It evaluates the CSI therein in comparison with its last CSI feedback to the transmitter, and, upon identifying a trend of severe CSI staleness, requests the transmitter to initiate a polling for *next* beamforming transmission. It conveys this request to the transmitter by setting the "Protected Frame" bit in the ACK for current data packet. The Protected Frame bit is an unused field in legacy 802.11 ACK packets.

For frequency-domain compression, AFC extracts per-subcarrier CSI from the 802.11 Long-Training-Field (LTF) preamble, which precedes the data portion of each packet, and is sent by each transmit antenna sequentially without beamforming. Based on the LTF, AFC designs an estimation algorithm to quantify the expected compression error and resulting capacity loss when several subcarriers are grouped to share the same CSI. The estimation of compression error induced by numerical quantization follows a similar procedure. With such estimations, AFC can select the combination of subcarrier grouping and quantization with the highest expected throughput, taking into account overhead, capacity loss as well as the time-domain feedback period. If this combination differs from the current grouping, the receiver again dictates the transmitter to reinitiate the polling for next beamforming packet. As in 802.11n/ac, information about the subcarrier group size or quantization bits are embedded in the receivers' CSI feedback packet.

For a MU-MIMO network, each receiver performs adaptation as if it forms a SU-MIMO link with the multi-antenna transmitter. Decoupling of their operations enables decentralized AFC and precludes complicated message exchanges between clients, which can be costly especially in large networks. We note, however, that the adaptation of quantization bits may be coupled among MU-MIMO receivers, and can lead to unfair power allocation. This problem will be elaborated in Sec. 4.3.2.

A unified metric for compression. The above adaptation mechanism entails a key question: *How does a certain level of compression affect throughput?* AFC proposes a unified metric, called *CNo* (compression noise), to evaluate the error introduced by each dimension of compression. CNo is obtained by inspecting the per-subcarrier CSI from the 802.11 LTF preamble. With CNo, AFC can instantaneously estimate the expected data-rate and weigh it against feedback overhead, to determine the optimal compression intensity. This evades the complexity of an intuitive search algorithm that attempts all combinations of compression intensities and leads to a huge search space, especially along

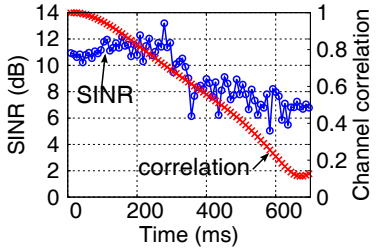

Figure 7: Channel correlation and link SINR change over time. The receiver is moving at walking speed.

the time dimension. In addition, CNo can be estimated separately for time, frequency and numerical values. Such isolation is important as it guarantees the total compression noise introduced by all dimensions is simply the sum of their CNo metrics.

In what follows, we detail AFC's adaptation mechanisms along three-dimensions, based on the CNo metric.

4.2 Time-domain adaptation

4.2.1 Is coherence time a good adaptation metric?

A channel's tolerance to time-domain compression strongly depends on channel stability. A widely used stability metric is coherence time, defined as the interval within which channel gain remains highly correlated. According to the 802.11ac recommendation [6], correlation between two instances of channel estimations, separated by T, is defined as:

$$\mathcal{K}(T) = \frac{L}{L-T} \cdot \frac{|\sum_{t=0}^{L-T-1} h(t)h^*(t+T)|}{\sum_{t=0}^{L-1} |h(t)||h^*(t)|} \quad (3)$$

where L is the total length of the sequence of sampled channel gains. Coherence time is defined as an instance $T_{0.5}$ when correlation drops to 0.5, *i.e.*, $T_{0.5} = \mathcal{K}^{-1}(0.5)$. Occasionally, $T_{0.9}$ is also used as a coherence time definition.

It is tempting to use coherence time as an update interval for CSI — CSI feedback can be initiated periodically, with a period equal to coherence time. All packets can share one CSI obtained in the beginning of each period. However, through extensive experiments, we found coherence time to be a poor gauge for predicting the effect of such CSI compression. In particular, the experiment in Fig. 7 shows that, even though channel correlation remains above 0.5 within a 400ms interval (corresponding to the $T_{0.5}$ definition for coherence time), SNR is already severely degraded before that (at around 260ms). On the other hand, the $T_{0.9}$ definition of coherence time is too conservative. Channel correlation is above 0.9 within 100ms, but SNR remains relatively stable within 200ms. Thus sending CSI at a peroid of 100ms incurs unnecessary overhead.

In addition, we found that this tradeoff manifests differently for different channel profiles (static, human moving around, or node mobility). Hence, no single definition of coherence time can be used as the period for time-domain compression.

An additional limitation of the coherence time metric is that the statistical correlation (3) is relevant only with a large number of channel gain samples, each corresponding to one packet. To adapt compression intensity, however, a responsive metric is needed that can be obtained from one or two packets. The CNo metric is designed to satisfy this requirement.

4.2.2 CNo metric for the time domain

Compression noise. In AFC, a receiver uses a Compression Noise (CNo) metric to overcome the limitations of conventional channel stability metric. CNo gauges how current CSI (at time t) differs from the last CSI feedback (at time t_0). Suppose there are K subcarriers whose CSI needs feedback and $H_k(t)$ is the channel gain for subcarrier k at time t. Due to channel variation, $H_k(t)$ can be considered as a noisy version of $H_k(t_0)$:

$$H_k(t) = H_k(t_0) + N_k(t) \quad (4)$$

We model the noise $N_k, (k \in K)$ as complex Gaussian random variables with a similar level of variance (*i.e.*, noise power). This is reasonable as the staleness of CSI applies equally to all subcarriers. Consequently, we can model the time-domain compression-induced noise power as:

$$\mathcal{N}_T(t) = \sum_{k=1}^{K} \Big| \big(H_k(t) - H_k(t_0)\big)\big(H_k(t) - H_k(t_0)\big)^* \Big|$$

This is exactly the CNo metric we use in time-domain. The receiver stores $H_k(t_0)$, and directly obtains $H_k(t)$, the per-subcarrier channel gain, from the 802.11 LTF preamble for each packet. Therefore, $\mathcal{N}_T(t)$ becomes an instantaneous gauge for time-domain compression error. Taking into account the signal power $P_r(t)$ and channel noise power $N_r(t)$, the expected SNR after compression becomes:

$$SNR(t) = \frac{P_r(t)}{N_r(t) + \mathcal{N}_T(t)} \quad (5)$$

An AFC receiver compares $SNR(t)$ with $SNR(t_0)$. If for any channel between a transmit antenna and a receive antenna, the difference is sufficient to cause data rate to drop from the current level to a lower level, then the receiver will request for CSI polling. Note, however, that channel fading may cause unpredictable SNR decrease/increase, which must be isolated in order to evaluate the true effect of time-domain compression.

Isolating fading effects. To solve this problem, the receiver also stores the nominal expected SNR (without compression noise) at t_0: $SNR'(t_0) = P_r(t_0)/N_r(t_0)$. It compares $SNR'(t_0)$ with $SNR'(t)$. If the difference exceeds the SNR threshold for boosting data rate to a higher level (threshold SNR_h^{th}), or degrading it to a lower level (threshold SNR_l^{th}), then the receiver will also request for a CSI polling. This helps the transmitter to select the bit-rate that fits the channel condition[5]. The SNR threshold for each data rate can be readily obtained from WiFi chip specifications (*e.g.*, [7]).

Note that a link may become idle for a long time. Without data transmission, the transmitter cannot receive any ACK or CSI update indication from the receiver, and is unaware of CSI staleness. AFC overcomes this problem by setting a maximum update interval T_m, which equals the average interval of the past 5 CSI update indication from the receiver. If the link becomes idle for more than T_m, the transmitter will initiate CSI polling without a request from the receiver.

Signal and noise power. So, how does AFC estimate the signal power $P_r(t)$ and noise power $N_r(t)$? We remark that such estimation should be isolated from the beamformed data portion of a packet, which mixes the channel gains from multiple transmit antennas to the receive antenna. Towards this end, we again leverage the LTF preamble which is sent by each transmit antenna sequentially. Fig. 8 illustrates the

[5]AFC's adaptation algorithm does not depend on the bit-rate adaptation mechanism.

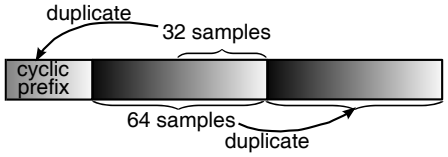

Figure 8: Time-domain structure of the LTF preamble in an 802.11 packet.

structure of an LTF preamble, which comprises two duplicated sequences (each containing 64 samples) and an OFDM cyclic prefix (32 samples). At the receiver side, it runs FFT over each sequence to estimate the frequency-domain channel distortion:

$$Y_k^1 = H_k X_k + N_k^1; \quad Y_k^2 = H_k X_k + N_k^2$$

X_k is a known symbol carried by subcarrier k. Although N_k^1 and N_k^2 are instances of two random Gaussian variables, they can be considered to have the same variance, which equals the noise power per-subcarrier. Thus the channel noise power over an entire bandwidth can be estimated as:

$$N_r(t) = \sum_{k=1}^{K} |(Y_k^2 - Y_k^1) \cdot (Y_k^2 - Y_k^1)^*| \quad (6)$$

Subtracting noise power from total received power, we obtain:

$$P_r(t) = \sum_{k=1}^{K} |Y_k^1|^2 - N_r(t) \quad (7)$$

Combating random phase offset. Ideally, if channel is stable, the receiver should experience a similar H_k for each packet. Due to the CSMA-based MAC layer, however, each transmitted packet has a random starting time and when modulated, will result in a random initial phase offset relative to the receiver. This causes a large $\mathcal{N}_T(t)$ even though the CSI does not change. We observe that transmit antennas on a MIMO radio share the same clock, and thus we use the phase of the differential channel H_k^A/H_k^B between transmit antenna A and B, instead of that of H_k directly when computing $\mathcal{N}_T(t)$, which removes the random phase offset.

4.3 Adaptive frequency-domain compression and numerical quantization

4.3.1 *Is coherence bandwidth a good adaptation metric?*

Coherence bandwidth is a classical metric for evaluating channel stability over frequency. Specific to 802.11 OFDM systems, it is defined as the range within which subcarriers' channel gain correlation is larger than 0.5 ($B_{0.5}$) or 0.9 ($B_{0.9}$). The correlation can be obtained in the same way as time correlation as in (3), except the separation is in frequency (number of subcarriers).

As channel gains of the subcarriers within $B_{0.5}$ or $B_{0.9}$ are highly correlated, one may wonder if they can be compressed and represented by a single subcarrier's CSI. The testbed experiments in Fig. 9 invalidate this perception. While $B_{0.9}$ is too conservative, $B_{0.5}$ is over-optimistic, incorporating more subcarriers than should be compressed, thus degrading capacity. Consistent with the experiments in Fig. 4, we found such effects to differ among different carrier frequencies and locations. In addition, correlation does not always decrease monotonically with subcarrier separation. Clearly, no single, static definition of coherence bandwidth can be used as a threshold for frequency-domain compression.

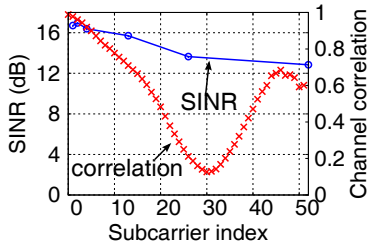

Figure 9: Channel correlation over frequency (in a lab environment), and SU-MIMO link capacity when K' subcarriers share the same CSI. PHY parameters are configured following 802.11 specification: 20MHz bandwidth, 52 data subcarriers (each occupying 312.5KHz).

4.3.2 *CNo metric for subcarrier grouping and numerical quantization*

We generalize the time-domain CNo metric to predict the link throughput for a certain level of frequency-domain compression and numerical quantization. Suppose $H_k(f_0, q_0)$ is the true CSI for subcarrier k without compression, which is obtained from the receiver's LTF preamble. $H_k(f, q)$ is the compressed CSI for subcarrier k with subcarrier group size f and q-bit quantization. All subcarriers inside each group share the CSI of the group's first subcarrier, following the 802.11 specification. Then, given a fixed quantization level q, the frequency domain CNo metric is:

$$\mathcal{N}_F(f) = \sum_{k=1}^{K} \left|\left(H_k(f,q) - H_k(f_0,q)\right)\left(H_k(f,q) - H_k(f_0,q)\right)^*\right|$$

Similarly, given a subcarrier group size f, the quantization noise with q-bit quantization is:

$$\mathcal{N}_Q(q) = \sum_{k=1}^{K} |(H_k(f,q) - H_k(f,q_0))(H_k(f,q) - H_k(f,q_0))^*|$$

Then, the receiver estimates the effective SNR with compression intensity f as $SNR'(f,q) = \frac{P_r}{N_r + \mathcal{N}_F(f) + \mathcal{N}_Q(q)}$. As P_r and N_r are obtained from packet preambles, they cannot reflect the power increase due to beamforming or noise increase due to multi-user cross-talk interference. Nevertheless, the receiver can estimate the true SNR based on the decoded SNR of the current packet, accounting for the SNR loss due to compression, *i.e.*,

$$SNR_{dB}(f,q) = decodedSNR_{dB} + SNR'_{dB}(f,q) - 10\log_{10}\frac{P_r}{N_r}$$

This can be mapped to data rate typically provided by WiFi vendors [7]. Then, we can estimate the increase of data packet transmission time, compared with the current compression intensity, say (f_1, q_1), as:

$$\Delta T_d(f,q) = D/Rate(SNR(f,q)) - D/Rate(SNR(f_1,q_1))$$

where D is the average data packet size (estimated by averaging over past 10 packets in AFC).

Owing to compression, the decrease of time in transmitting CSI, amortized over each packet, is:

$$\Delta T_o(f,q) = Kq_1M_t(f_1R_0T_p)^{-1} - KqM_t(fR_0T_p)^{-1}$$

where M_t is the number of transmit antennas, and R_0 the data rate of feedback packet (default to 6Mbps in 802.11). T_p is the number of recent packets that shared one CSI over time, which reflects the current time-domain compression intensity.

AFC compares $\Delta T_o(f,q)$ with $\Delta T_d(f,q)$. A positive difference implies benefit for changing the compression intensity. This comparison is made for each possible combination of subcarrier group size and quantization level (in total there are only 12 such combinations in 802.11 as introduced in Sec. 2), and the receiver will select the combination with largest $\Delta T_o(f,q) - \Delta T_d(f,q)$. If it is positive and differs from the current one, the receiver switches to this new compression level and notifies the transmitter. Again, the above estimation is performed for each transmit antenna separately. CSI feedback will be triggered whenever one antenna passes the test.

Isolating different users' CSI quantization. A unique problem in MU-MIMO is that different users' CSI quantization may affect each other's performance. When users choose different quantization levels, their CSI will be scaled disproportionally. Suppose user A and B have a similar level of channel gain. If user A adopts 8-bit quantization whereas user B adopts 4-bit, then A's CSI feedback will have $2^4 = 16$ times as high magnitude as B's, causing the transmitter to think A has much better channel quality. Since zero-forcing precoding implicitly projects the same received power towards both receivers (Sec. 2), the transmitter will allocate only $\frac{1}{\sqrt{16}} = \frac{1}{4}$ as high power to user A as the case without quantization, resulting in unfair capacity loss for A.

In AFC, the transmitter solves this problem by normalizing the real and imaginary components with the maximum possible value ($2^{q-1}-1$ for q-bit quantization), for each user respectively. This prevents the unfair CSI scaling and thus isolates the quantization choices of different users. Note that AFC's adaptation algorithm is immune to such an effect, as it only leverages the non-beamformed LTF preamble.

4.4 3-D joint adaptation

Algorithm 1 summaries AFC's adaptation over all three dimensions. Note that the CNo metric serves as a unified function for estimating compression noise for each dimension. It naturally isolates the effects of each dimension, and reduces the search space for throughput-optimal compression. It is also decentralized with respect to each user, who only needs to evaluate CNo between itself and the transmit antennas. In addition, unlike traditional metrics (*e.g.*, coherence time), it can be obtained instantaneously after receiving each packet. The complexity of AFC's adaptation depends on the number of subcarriers and transmit antennas, all of which are limited in 802.11n/ac. Thus AFC involves a constant computation time and is amenable for implementation in real WiFi drivers.

Algorithm 1 3-D cross-layer adaptation in AFC.

1. **Transmitter module:**
2. **if** "Protected Frame" bit in ACK equals 1 **or** link idle time exceeds T_m
3. Initiate CSI polling and request for next packet
4. Run SU-/MU-MIMO beamforming with latest CSI
5. **endif**
6.
7. **Receiver module:**
8. /* Time-domain adaptation */
9. Compute CNo metric $\mathcal{N}_T(t)$ and $SNR(t)$
10. **if** $SNR(t) - SNR(t_0) > SNR_l^{th}$ **or** $SNR'(t) - SNR'(t_0) > SNR_h^{th}$ **or** $SNR'(t_0) - SNR'(t) > SNR_l^{th}$
11. Request CSI update; update T_p
12. **endif**
13. /* Adaptive subcarrier grouping and CSI quantization*/
14. Compute CNo metric $\mathcal{N}_F(f)$ and $\mathcal{N}_Q(q)$
15. Compute expected TX time increase $\Delta T_d(f,q)$ and overhead time decrease $\Delta T_o(f,q)$
16. Search for $(f^*, q^*) = \arg\max_{f,q} \Delta T_o(f,q) - \Delta T_d(f,q)$
17. **if** $\Delta T_o(f^*,q^*) - \Delta T_d(f^*,q^*) > 0$
18. Switch to new subcarrier group size f^* and quantization level q^*
19. Request CSI update; **continue** to next transmission.
20. **endif**

5. IMPLEMENTATION

5.1 Testbed implementation

We have implemented a full-fledged 802.11ac-compatible MU-MIMO OFDM system on the WARP [8] platform, based on the WARPLab driver which provides interfaces to the WARP software radio. Fig. 10 illustrates the main components in our system. The transmitter uses linear precoding (ZFBF, Sec. 2) to pre-cancel inter-user interference. Precoded data symbols are divided into M_t streams (equal to the number of transmit antennas), each subject to OFDM modulation. The modulated data symbols are prepended with the 802.11 LTF, and STF (a preamble for synchronization). Note the SU-MIMO is a special case of MU-MIMO implementation, as introduced in Sec. 2.

The receiver uses an auto-correlation algorithm to detect the STF and identify exact starting time of each packet. It then leverages the LTF to estimate the channel matrix between itself and each transmit antenna. This channel matrix (CSI) will be compressed by the receiver using AFC and then fed back to the transmitter when needed. As inter-user interference is pre-cancelled, the receiver is able to decode the data portion of the packet using a standard OFDM demodulator.

The 3-D cross-layer adaptation algorithm is implemented at the receiver side following the description in Sec. 4. It extensively leverages the LTF to compute CNo along three dimensions, and decides if an adjustment of compression intensity is needed and in which dimensions. As the round-trip time between the PC (the central MAC/PHY processor running AFC and MU-MIMO OFDM modulation/demodulation) is several orders of magnitude longer than real WiFi hardware, we do not directly implement real-time control messages such as ACK, NDP and CSI feedback. Instead, as all WARP nodes in our testbed share the same central processor, control messages are directly realized as function calls. The exact transmission duration and inter-packet spacing (defined in 802.11) are implemented using a virtual timer. In addition, all MIMO data packets are sent through real channels by the WARP radios, and polling packets are initiated if AFC dictates a CSI refresh.

5.2 Trace-driven emulation

Through benchmark tests, we found it takes 127ms for a pair of WARP radios to finish an entire SU-MIMO transmission involving CSI polling, precoding, data transmission and data decoding. The latency is mainly caused by the hardware interface between WARP and its PC host. Such hardware latency is incompatible with real WiFi radios, and can-

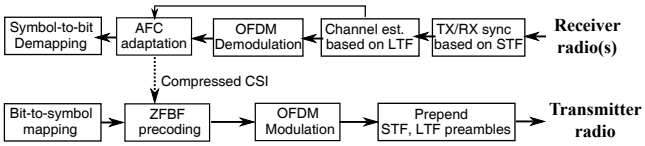

Figure 10: Implementation of an 802.11-compatible MU-/SU-MIMO OFDM system.

not faithfully reproduce AFC's behavior in dynamic/mobile scenarios. To overcome this limitation, we adopt a trace-driven emulation approach, which is inspired by [9] but harnesses WARP's flexibility to collect raw channel traces.

We fill a WARP transmitter's transmit buffer with an 802.11 MIMO preamble (containing STF and LTF sequences), and configure it to repeatedly send the preamble over the air. The receivers log the raw digital samples to a file, and performs packet detection and channel estimation offline to obtain a sequence of channel traces. With this setting, we are able to obtain fine-grained sampling of the MIMO channel every 8.7ms, which is sufficient for evaluating mobile channels with walking speed [10].

To evaluate the feedback compression algorithms, we replace the WARPLab interface to the WARP radios with an interface to an emulated channel, whose time variation follows the collected traces. The MU-/SU-MIMO OFDM and AFC module that we implemented can run directly over this emulated channel. When a packet's arriving time does not exactly match any instances of the channel trace, the closest instance is used to represent its channel distortion. We will show in Sec. 6 that the trace-driven emulator can faithfully reproduce the results from realtime experiments.

6. EVALUATION

In this section, we first evaluation AFC's adaptation modules through micro-benchmark experiments. Then we integrate all the modules and validate the system-level performance under various channel conditions.

6.1 Micro-benchmark evaluation

6.1.1 *Validation of trace-driven emulation.*

Part of our experiments are conducted under mobile/dynamic channel profiles, and rely on trace-driven emulation. Thus, we first validate the accuracy of such trace-driven emulation, using real-time runs as a benchmark.

	T_1	T_c	Freq 1	Freq 52	Qbits 2	Qbits 8
Trace	20.206	20.041	20.262	16.131	18.662	20.315
Real	20.142	20.074	19.825	15.148	17.599	20.066
Error	0.32%	0.16%	2.20%	6.49%	6.04%	1.24%

Table 1: Trace-driven emulation can faithfully reproduce the SINR results from realtime experiments. T_1 and T_c denote per-packet and per-coherence-time CSI feedback, respectively. Freq denotes subcarrier group size 1 and 52. Qbits denote quantization bits.

Following the steps in Sec. 5.2, we first run SU-MIMO transmission between a 2-antenna WARP transmitter and single antenna receiver, while logging the per-packet channel matrices between them. Then we replay the transmissions

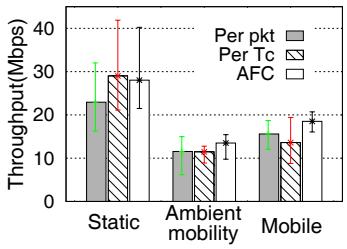

Figure 11: Time-domain adaptation: SU-MIMO.

in the emulator based on the traces. Table 1 compares the results from real-time runs and trace-driven emulation. We create 6 traces, each lasting 1 minute and representing the case with or without compression (or with minimum compression) along three dimensions. For the majority of cases, relative error of emulation is well below 3%. When SINR is low, the relative error tends to be magnified (to around 6%). However, the absolute error remains below 1.1dB. Therefore, the trace-driven emulator can reproduce real-time experiments with a sufficient level of accuracy.

6.1.2 *Time-domain adaptation*

The effectiveness of AFC's time-domain compression and adaptation depends on channel stability. Thus we evaluate this component under three representative channel profiles: static, ambient mobility (human moving around the nodes) and node mobility (node moving at walking speed). All experiments run over the 2.4GHz WiFi channel 14, which is unused by nearby wireless devices. The static case runs directly on the WARP testbed. The other two cases are based on trace-driven emulation, since WARP's latency may exceed the time-resolution needed for real-time CSI feedback.

We compare AFC against two alternative CSI feedback schemes: per-packet feedback and Per-T_c feedback. The former is the default scheme in 802.11n/ac. In the latter scheme, we first collect channel traces, compute the coherence time $T_{0.5}$, and then use $T_{0.5}$ as the feedback period when replaying the traces.

SU-MIMO network. In a typical small or medium sized network, contention overhead is much smaller compared with data transmission. Thus, without loss of generality, we first consider a single receiver SU-MIMO network[6], where the transmitter has 2 antennas and receiver 1 antenna. Fig. 11 plots the resulting network throughput.

In static case, channel variation is negligible, and thus per-packet CSI feedback incurs huge overhead. AFC only needs to opportunistically feed back CSI without affecting link capacity noticeably. Its mean throughput is 25.4% higher than that of per-packet feedback. Per-T_c feedback results in a similar level of throughput as AFC.

With ambient mobility, per-packet feedback degrades even with per-T_c feedback in terms of throughput, as the former maintains link capacity whereas the latter attempts to reduce feedback overhead. In contrast, AFC strikes a balance between these two objectives, improving average throughput by 17% over the other two schemes.

When receiver is mobile, Per-T_c feedback underestimates channel variations, leading to even lower throughput than per-packet feedback. On the other hand, per-packet feedback incurs more overhead. AFC's time-domain adapta-

[6]Even for a single-link WLAN, contention backoff is required after each transmission attempt [4].

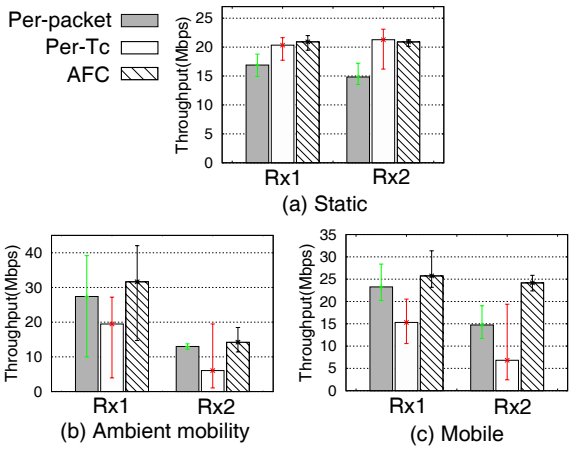

Figure 12: Time-domain adaptation: MU-MIMO.

tion mechanism can effectively estimate the instances when CSI feedback is needed. Its throughput is 18.2% and 38.5% higher than per-T_c and per-packet, respectively.

MU-MIMO network. We further evaluate AFC in a MU-MIMO network with a 2-antenna AP and 2 single-antenna receivers. Fig. 12 plots the receiver's throughput under different channel profiles and feedback schemes. Similar to SU-MIMO, per-T_c achieves a comparable level of throughput with AFC. However, for per-packet feedback, since the overhead doubles compared with SU-MIMO, the throughput loss is substantial. In particular, AFC can achieve 39.7% higher throughput over per-packet — a much higher gain compared with SU-MIMO case.

Under channel dynamics, MU-MIMO is expected to be more sensitive to CSI feedback errors. Fig. 12 testifies this intuition. With ambient mobility, per-T_c induces long feedback delay and large CSI error, resulting in significant throughput loss. AFC can achieve 61.5% and 123% higher throughput, for receiver 1 and 2, respectively.

AFC's performance gain is most remarkable in mobile case (Fig. 12(c)). It achieves 69.1% and 269% higher throughput for the two receivers, compared with per-T_c; and a 13.2% and 71.3% gain compared with per-packet feedback.

In summary, none of the static CSI feedback mechanisms are appropriate for all channel profiles. On the other hand, by triggering CSI updates only when needed, AFC can achieve up to multi-folds of throughput gain over static schemes.

6.1.3 Frequency-domain adaptation

We proceed to evaluate whether AFC can effectively adapt to an appropriate level of compression for the frequency domain, *i.e.*, the subcarrier group size. Our investigation of channel traces in a lab environment reveals that even for the same spectrum band, coherence bandwidth may change, *e.g.*, due to minor change of location or antenna orientation. Thus we create frequency domain dynamics by running multiple experiments with minor adjustment of transmitter's locations. We isolate the effect of the other two adaptation dimensions by assuming default configuration in 802.11n/ac (per-packet CSI feedback and 8-bit quantization).

Fig. 13 plots the throughput of AFC compared with two alternative schemes with fixed subcarrier group size 1 and 52, respectively. For SU-MIMO, AFC's throughput is 19.8% higher than Size1, implying that a fixed subcarrier group size of 1 underutilizes the channel correlation between subcarriers, which could have been harnessed to compress the CSI. In our lab environment, Size52 achieves a similar level of throughput with AFC.

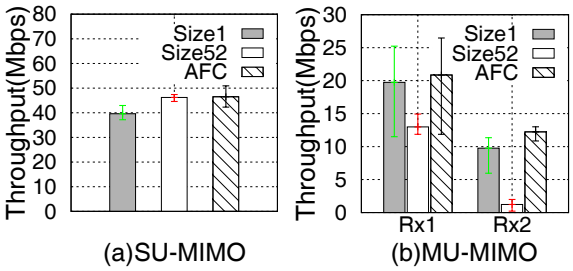

Figure 13: Frequency-domain adaptation for (a) SU-MIMO and (b) MU-MIMO. Error bars denote maximum and minimum among 5 consecutive experiments, unless noted otherwise.

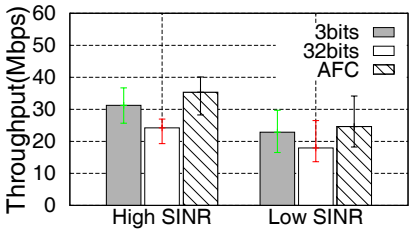

Figure 14: Adaptive quantization: SU-MIMO.

However, a MU-MIMO network is more sensitive to CSI compression errors, since severe inter-receiver interference can occur when CSI is noisy. In particular, Fig. 13(b) shows that for one receiver (Rx2), AFC's throughput can be 5.2× that of Size52, and 1.36× for Rx1. In this case, a fixed subcarrier group size of 1 outperforms Size52.

Therefore, from frequency domain perspective, static subcarrier grouping can be either very conservative or aggressive. AFC can effectively strike a balance, selecting the best compression level to optimize the network throughput.

6.1.4 Adaptive quantization

To evaluate AFC's adaptive quantization mechanism, we compare it against static quantization algorithms with aggressive (3-bit) and conservative (32-bit, essentially no compression) quantization algorithms. Accordingly, we allow AFC to choose from these two quantization levels along with the default options in 802.11n/ac (4, 5 6 and 8 bits). Fig. 14 shows the experimental results for a SU-MIMO network. It can be seen that for both high and low SINR[7] cases, the saving of feedback overhead dominates the capacity loss, even with a very aggressive 3-bit quantization. However, there is still sufficient space for AFC to optimize the choice of quantization levels. An examination of experimental results reveals that AFC tends to choose the quantization level of 6 bits, which is close to 3-bit but avoids the capacity loss due to excessive compression.

In addition, as we noted above, in high SINR case, closed-loop MIMO transmission is more sensitive to compression, hence aggressive compression tends to suffer more from capacity loss. Indeed, Fig. 14 shows that AFC improves throughput by 15.1% over 3-bit compression in high-SNR region, and

[7]We adjust link distance to ensure the receivers' SINR level satisfy the requirement of the minimum level of modulation in 802.11, which is 9dB for BPSK modulation [7].

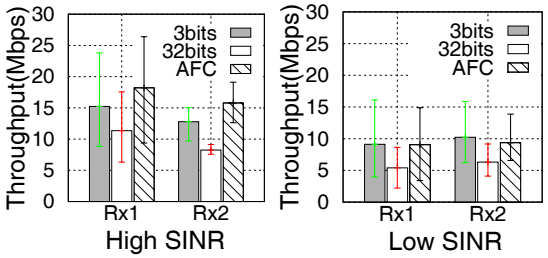

Figure 15: Adaptive quantization: MU-MIMO.

4.5% in low-SINR region. Owing to its intelligent overhead reduction, performance gain over fixed 32-bit reduction is more prominent (52.2% and 28.3% for high and low-SINR case, respectively).

Experimental results from a MU-MIMO network (Fig. 15) show a similar trend. In particular, for low-SINR case, channel noise dominates CSI error, and thus 3-bit quantization achieves comparable throughput as AFC. But in high SINR region, AFC achieves 18.5% higher throughput. The gain is even higher than in SU-MIMO, because aggressive quantization amplifies the effects of inter-receiver interference.

6.2 Field test of AFC

In this section, we integrate all three components of AFC and evaluate its network performance on our WARP radio testbed.

6.2.1 AFC in action

We first check whether AFC can react to changes in a real wireless link with unpredictable dynamics. To this end, we evaluate a 2-receiver MU-MIMO network, and vary the transmitter's location and link distance, such that a variety of channel profiles can be generated for both receivers. We collect the per-packet channel traces during this process and then replay them in our emulator.

Fig. 16 plots the throughput variation of AFC over time, in contrast with the legacy configuration in 802.11ac (per-packet feedback, subcarrier group size 1, 8-bit quantization). Throughput is calculated for each inter-packet arrival interval. The results show that AFC can react instantaneously to the change of channel state, and effectively adapt its compression levels accordingly. It consistently achieves higher throughput than 802.11ac under any channel dynamics. The throughput gain ranges from 1.4× to 2.2× in general, and varies depending on the channel profile and users. Occasionally, the fixed configuration in 802.11ac may work as well as AFC when it is appropriate for the channel state.

6.2.2 Other factors

As mentioned in Sec. 3, besides time, frequency and quantization, there are a number of other factors that determine the CSI overhead, which may in turn affect AFC's choice of compression. Below we use testbed experiments to verify such effects.

We first evaluate packet length, which affects relative overhead of CSI feedback and hence throughput. To isolate the effects of bit-rate change, we vary packet duration instead of size. Experiments in a SU-MIMO network (Fig. 17) show that for shorter packet durations, AFC achieves higher gain (*e.g.*, 2.14 × for 100 μs). In such cases, balance between overhead and capacity becomes more important, which cannot be handled by the legacy scheme.

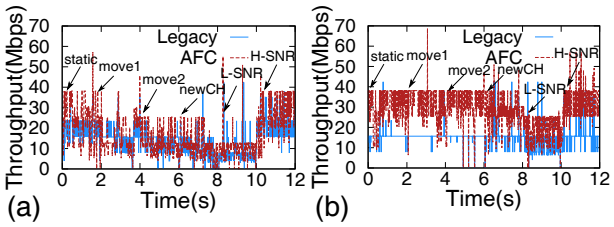

Figure 16: Field test of AFC in a MU-MIMO network. "newCH" indicates a change of spectrum band. "L-SNR" and "H-SNR" denote low-SNR and high-SNR scenarios created by adjusting transmit power.

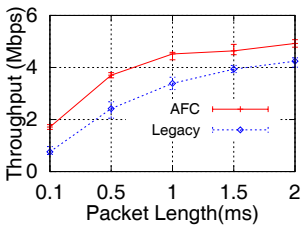

Figure 17: Effects of packet length on MIMO network throughput.

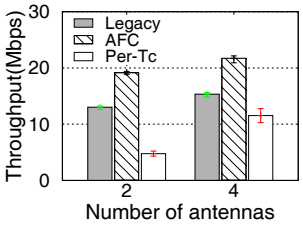

Figure 18: SU-MIMO network throughput with 2 and 4 transmit antennas.

Fig. 18 further shows how this balance manifests under different number of antennas. The experiment is done in a lab environment with receiver moving at walking speed. AFC's throughput gain is around 47% higher than legacy for both 2 and 4 antenna cases. Under 2 antenna case, Per-T_c feedback underestimates channel variation, resulting in 72% lower throughput compared with AFC. With 4 antennas, feedback overhead plays a greater role, and thus Per-T_c feedback enjoys throughput improvement due to its aggressive time-domain compression, but its throughput is still 45% lower than AFC.

7. RELATED WORK

CSI feedback has been a critical issue in broadband cellular networks such as LTE [11]. For CSI quantization, early generations of LTE adopt codebook based approach, which prescribes a set of precoding vectors, such that the receiver only needs to feed back an index of the best precoding vector [12]. Latest generation of LTE (Release-10) proposes quantized CSI feedback, which is shown (via simulation) to achieve comparable performance with the same number of CSI bits [13]. But even for LTE, the adaptive selection of quantization levels or feedback delay remains an open problem. There exists a vast literature of work on limited feedback MIMO communications (see [11] for a comprehensive survey), which focuses on the design of effective CSI quantization mechanisms to restrict the feedback overhead. In [14], Huang *et al.* analyzed the effect of time-domain compression, based on a theoretical model of channel correlation over time. Pohl *et al.* [15] simulated the SNR and BER of MIMO communication systems under Rayleigh fading model. However, a systematic approach is still lacking that adapts various quantization and compression mechanisms in realistic LTE channel conditions.

Relatively less work has been devoted to the CSI feedback mechanism in wireless LANs. In [16], Sun *et al.* simulated the 802.11n SU-MIMO performance in time-varying

and frequency-selective channel conditions. Crepaldi *et al.* [17] proposed a sampling algorithm that can efficiently estimate the channel matrix of a MIMO link. To our knowledge, there does not exist any experimental work that addresses the tradeoff between CSI compression and capacity loss for WLANs.

Recently, there has been a number of software-radio based prototypes of multi-user MIMO [18] or distributed MIMO networks [19], which verified the practicality of theoretical MIMO communication algorithms, but ignored the CSI feedback overhead. In [20], Shepard *et al.* built a massive multi-user MIMO system, based on implicit CSI feedback — an alternative feedback mechanism defined in 802.11n, which leverages channel reciprocity of uplink/downlink to reduce CSI overhead. Implicit beamforming involves sophisticated calibration that is not amenable for handheld WiFi devices [21, 22]. In addition, it cannot be used in many practical wireless transceivers that have more receive antennas than transmit antennas [21, 23]. The new generation of 802.11ac has discarded implicit beamforming, leaving explicit CSI as the only option for CSI feedback.

8. CONCLUSION

In this paper, we have conducted a comprehensive measurement study that characterizes the tradeoff between feedback compression and capacity loss in SU-/MU-MIMO wireless networks. We then address this tradeoff by proposing AFC, a simple mechanism that adapts compression intensity to optimize network throughput. AFC's adaptation is steered by CNo, a unified metric that can be extracted from 802.11 packets' preambles, and used to estimate the noise effects due to compression in time, frequency and numerical values. We implemented AFC on the WARP software radio testbed and validated its performance under various channel dynamics and network scenarios. We believe AFC can be a general mechanism for other non-802.11 MIMO networks, such as those based on interference alignment [24]. Such extension is left for our future exploration.

9. REFERENCES

[1] F. Boccardi and H. Huang, "Zero-Forcing Precoding for the MIMO Broadcast Channel under Per-Antenna Power Constraints," in *IEEE 7th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2006.*, 2006.

[2] D. Gesbert, M. Kountouris, R. Heath, C.-B. Chae, and T. Salzer, "Shifting the MIMO Paradigm," *Signal Processing Magazine, IEEE*, vol. 24, no. 5, 2007.

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

[4] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std. 802.11n*, 2009.

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

[6] G. Breit, "Coherence Time Measurement for TGac Channel Model," *IEEE 802.11-09/1173r1*, 2010.

[7] Cisco Systems Inc., "Wireless Mesh Access Points, Design and Deployment Guide," *Release 7.3*, 2012.

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

[9] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery From Wireless Channel Measurements," in *Proc. of ACM SIGCOMM*, 2010.

[10] R. Kudo, K. Ishihara, and Y. Takatori, "Measured Channel Variation and Coherence Time in NTT Lab," *IEEE 802.11-10/0087r0*, 2010.

[11] D. Love, R. Heath, V. Lau, D. Gesbert, B. Rao, and M. Andrews, "An Overview of Limited Feedback in Wireless Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 8, 2008.

[12] D. Love, R. Heath, W. Santipach, and M. Honig, "What is the Value of Limited Feedback for MIMO Channels?" *IEEE Communications Magazine*, vol. 42, no. 10, 2004.

[13] P. Frank, A. Muller, and J. Speidel, "Fair Performance Comparison Between CQI- and CSI-based MU-MIMO for the LTE Downlink," in *European Wireless Conference (EW)*, 2010.

[14] K. Huang, R. Heath, and J. Andrews, "Limited Feedback Beamforming Over Temporally-Correlated Channels," *Signal Processing, IEEE Transactions on*, vol. 57, no. 5.

[15] V. Pohl, P. Nguyen, V. Jungnickel, and C. Von Helmolt, "How Often Channel Estimation is Needed in MIMO Systems," in *Proc. of IEEE GlobeCom*, 2004.

[16] X. Sun, L. J. Cimini Jr., L. J. Greenstein, D. S. Chan, and B. Douglas, "Performance of Quantized Feedback Beamforming in MIMO-OFDM Links over Time-Varying, Frequency-Selective Channels," in *Proc. of IEEE MILCOM*, 2007.

[17] R. Crepaldi, J. Lee, R. H. Etkin, S.-J. Lee, and R. Kravets, "CSI-SF: Estimating Wireless Channel State Using CSI Sampling & Fusion," in *Proc. of IEEE INFOCOM*, 2012.

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

[19] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Achieving High Data Rates in a Distributed MIMO System," in *Proc. of ACM MobiCom*, 2012.

[20] 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.

[21] H. Zhang et al., "802.11ac Explicit Sounding and Feedback," *IEEE 802.11-10/1105r0*, 2010.

[22] Ruckus Wireless, Inc., "All Beamforming Solutions Are Not Equal — Discussion on the Topic of Beamforming," 2013.

[23] D. Halperin, "Simplifying the Configuration of 802.11 Wireless Networks with Effective SNR," *CoRR*, vol. abs/1301.6644, 2013.

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