Bringing Mobility-Awareness to WLANs using PHY Layer Information

Li Sun
University at Buffalo, SUNY
lsun3@buffalo.edu

Souvik Sen
HP Labs
souvik.sen@hp.com

Dimitrios Koutsonikolas
University at Buffalo, SUNY
dimitrio@buffalo.edu

ABSTRACT

With the proliferation of smartphones and tablets, mobile devices are soon becoming a preferred medium of Internet access in Wireless LANs (WLANs). Due to their smaller form factor, these truly mobile devices allow the users to access the wireless networks while undergoing different types of mobility, posing new challenges to wireless protocols. Current history-based protocols that maximize performance in static settings do not work well in mobile settings where wireless conditions change rapidly. Thus, today's WLANs need to be able to determine the type of the client's mobility and employ appropriate strategies in order to sustain high performance. While previous work tried to detect mobility using hints from sensors available in today's mobile devices, in this work, we demonstrate how different mobility modes can be distinguished by using physical layer information – Channel State Information (CSI) and Time-of-Flight (ToF) – available at commodity APs, with no modifications on the client side. In addition, we demonstrate how fine-grained mobility determination can be exploited to improve performance of client roaming, rate control, frame aggregation, and MIMO beamforming. Our testbed experiments show that our mobility classification algorithm achieves more than 92% accuracy in a variety of scenarios, and the combined throughput gain of all four mobility-aware protocols over their mobility-oblivious counterparts can be more than 100%.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

WLAN; Mobility; Channel State Information; Time-of-Flight; Client Roaming; Rate Adaptation; Frame Aggregation; Beamforming

1. INTRODUCTION

Wireless LANs (WLANs) were previously dominated by static network users with devices such as laptops and notebooks. However, with the proliferation of smartphones, tablets, and the advent of the BYOD phenomenon, mobile devices are soon becoming the preferred medium of Internet access. Due to their smaller form factor, these truly mobile devices allow the users to access the wireless network while undergoing different forms of mobility. Thus, unlike previously when WLANs used to mainly serve static clients, today they need to be optimized for mobile clients as well.

Client mobility poses difficult problems to the WLAN protocols. In static scenarios, the wireless channel remains stable and hence wireless protocols can refer to the past transmission history, carefully adapting themselves to avoid any failures and optimize performance. In contrast, during mobility, wireless conditions may change frequently requiring protocols to be agile and apply different strategies. Designing such strategies is not trivial because they depend on the actual type of mobility demonstrated by the client. E.g., the length of past history that a wireless protocol can refer to depends on the intensity of the mobility of the client. If the client is walking *towards* her access point (AP), the wireless channel is likely to become better and hence transmission bit-rate control can be more aggressive. Whereas if the client is walking *away* from her AP, bit-rate control needs to be conservative. The WLAN should try to move the client to a better AP only when the client is moving away from her AP and should not consider roaming for static clients because it is unlikely that a better AP will be discovered. Likewise, mechanisms to obtain higher throughput in modern 802.11n/ac WLANs, such as frame aggregation, beamforming, multiuser-MIMO (MU-MIMO), require different optimizations based on the intensity of mobility of the client. Therefore, unless the WLAN protocols can somehow determine the type of the client's mobility and thereafter employ appropriate strategies, it is unlikely that they will be able to sustain high performance.

Today's WLAN protocols are unable to distinguish between different types of client mobility that a wireless client may undergo. They generally try to sustain performance by applying heuristics to reduce wireless errors. If the number of errors is high, current protocols try to improve performance by employing aggressive strategies. On the other hand, if the number of errors is low, they try to maintain reliability by being more conservative. Such a general framework is slow to adapt to client mobility.

Recognizing the difficulty to detect mobility using the client's RSSI values, recent work [1] tried to detect mobility using the accelerometer in a mobile phone. While this technique can broadly classify between stationary and mobile devices, it has a few disadvantages. First, the use of client's on-board sensor means that mobility detection can only be performed by the mobile client. This requires changes at both the client and the AP because, to benefit download traffic, the protocol is required to communicate the device's mobility state to the AP. Second, this technique can be implemented only on client devices that have an accelerometer, and it requires the sensor to be on consuming battery life. Rather than changing the client, which is a more difficult proposition, we demonstrate how different mobility modes can be distinguished by using physical (PHY) layer information available at commodity APs.

In our study we identify four broad categories of client mobility. If the client is stationary, it can be in the *static mobility mode* when there are no significant environmental changes affecting the channel between the AP and the client. A static client may also be in the *environmental mobility mode* when the channel changes due to external movements. Of course the wireless client itself may be moving – a mobility mode that we call *device mobility*. The wireless client may experience different speeds under device mobility. However, we identify two broad and dominant categories of device mobility in WLANs. First, the user may slowly move the device although she is stationary or her movement is confined within a small area. E.g., the user may be attending a VoIP call over WiFi and a little movement of her head may displace her smartphone. The user may be playing an online multi-user game where the game controls require her to shake the phone or tablet frequently. She may be roaming around within her office or cubicle while watching a streaming video or listening to streaming music. We define that the client is under *micro-mobility* if it is under device-mobility but its location is confined within a small area. On the other hand, device mobility may also cause the client to change its location as its user walks from one location to another. In such scenarios, we classify the client to be under *macro-mobility*.

We utilize Channel State Information (CSI) and Time-of-Flight (ToF) values available from off-the-shelf HP MSM APs to determine the client's mobility mode. CSI captures the fine-grained characteristics of the wireless channel. We show that the temporal changes in the CSI can discriminate between static, environmental, and device mobility. However the CSI value may change similarly for micro and macro mobility scenarios. To distinguish between these two device mobility modes, we find that the client's distance from the AP changes significantly under macro-mobility, whereas under micro-mobility her distance seldom changes. The ToF can reliably indicate the distance between the client and the AP because it captures the round trip propagation time between the two. The increasing vs. decreasing trend of the client's distance can further indicate her relative heading with respect to the AP, whether the client is *moving away* from or *moving towards* the AP.

In addition to mobility determination using PHY layer information, we make the following contributions:

- **Client roaming:** In section 3, we demonstrate how the knowledge of the mobile client's relative heading can be exploited to improve client roaming in WLANs. By encour-
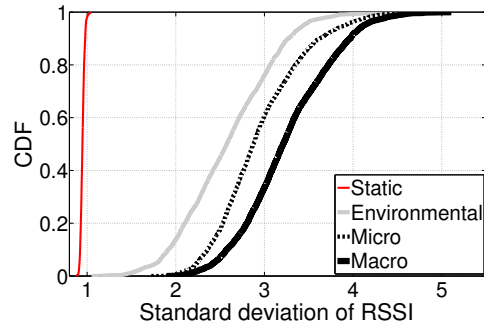

Figure 1: CDF of standard deviation of RSSI computed every 5 seconds, for various types of mobility.

aging the client to roam when it is moving away from its current AP, our protocol improves performance by up to 40% in our testbed experiments.

- **Rate control:** In section 4, we show how mobility-awareness can further improve an existing frame-based rate control scheme. Mobility hints can help the rate control scheme determine when it can aggressively increase rate or whether past history should be accounted for while choosing the bit-rate. Testbed experiments show that mobility-aware rate control can improve performance by up to 30%.

- **Frame Aggregation:** In section 5, we demonstrate that the frame aggregation size needs to be adapted depending on the client's mobility mode. Our testbed experimentation shows that, when compared with schemes using default aggregation size, our adaptive scheme can improve throughput by up to 30%.

- **Beamforming and MU-MIMO:** Beamforming and MU-MIMO require frequent and expensive CSI feedback from the client. In section 6, we show that the frequency of CSI feedback depends on the mobility mode of the client. By adapting the CSI feedback frequency based on client's mobility, we demonstrate that performance can be improved by up to 70% in testbed experiments.

2. MOBILITY CLASSIFICATION

This section presents our design and algorithms for mobility classification using PHY layer information. We begin with a description of our experimental setting using commodity APs and smartphones.

2.1 Experimental Setup

We use HP MSM 460 APs with Atheros AR9390 chipset and Samsung Galaxy S5 smartphones in our experiments. We tuned the AP at 5.805GHz using a 40MHz channel width and the 802.11n protocol. The MSM AP is equipped with 3 transmit antennas while the Samsung smartphone has 2 antennas. We sent regular data packets from the AP and collected CSI, ToF, and Received Signal Strength Indicator (RSSI) information from the acknowledgment (ACK) sent by the client. We ran four different experiments to analyze different classes of mobility. First, we evaluated static mobility by placing the smartphone in our lab with several people present but not moving. Second, for environmental mobility, we experimented in a cafeteria during lunch hours where the smartphone was static on a table. Third, for micro-mobility, we picked up the phone and moved it around within a meter
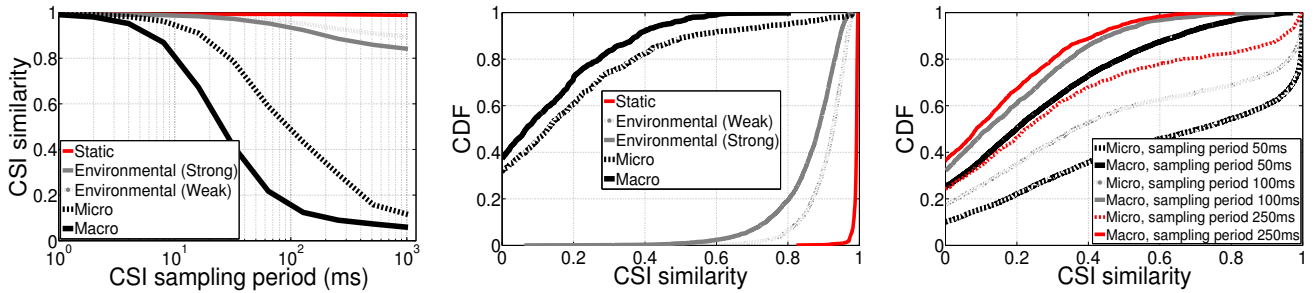
Figure 2: **(a) Variation of CSI similarity values over time. CDF of CSI similarity values (b) for various mobility modes, (c) for device mobility at different CSI sampling frequencies.**

of its location, using natural gestures. Lastly, for evaluating client macro-mobility, we walked naturally with the phone in our hand or inside the pocket. Our goal is to develop simple classification schemes that can distinguish among the above four scenarios.

2.2 Classifying Mobility using RSSI

We explore the possibility of classifying the mobility mode of the client solely by relying on the PHY layer information available from commodity WiFi APs. One possibility can be to utilize the RSSI of the client. In our experiments, we found that RSSI is quite stable in static scenarios in a quiet environment (Figure 1). However, RSSI is susceptible to any changes in the environment [1]. Often, the RSSI variation under environmental mobility is higher than the observed variation under device mobility (Figure 1). Therefore, we conclude that it is difficult to distinguish between environmental and device mobility solely based on RSSI. We will show next how CSI can conclusively classify these two types of mobility.

2.3 Classifying Mobility using CSI

Wireless is a broadcast medium. Any transmitted signal from the client undergoes reflections and arrives along multiple paths (multipath) at the AP. The wireless channel between the client and the AP is expected to vary under environmental or device mobility. This is because the fine-grained multipath structure may change if there are moving objects in the environment, or the device itself moves. These fine-grained variations of the wireless channel cannot be captured by RSSI because it only captures an aggregate indicator of all the multipath components. However, we found that the same variations can be reliably detected by CSI. CSI captures the fine-grained multipath characteristics between the client and the AP in the frequency domain. The Atheros 9390 chipset can export the CSI of a received packet. On a 20 MHz channel, CSI is reported as a matrix of complex numbers representing the channel gain for each of the 52 subcarriers and every transmit-receive antenna pair at the AP and the client. We track the change in the CSI values to detect any variations of the wireless channel. We define the *similarity* between CSI samples, csi_t and $csi_{t+\tau}$ collected at two different time instances t and $t+\tau$ as:

$$S(csi_t, csi_{t+\tau}) = \frac{\sum\limits_{i=1}^{52}\left(|csi_t^i| - \overline{|csi_t|}\right)\left(|csi_{t+\tau}^i| - \overline{|csi_{t+\tau}|}\right)}{\sqrt{\sum\limits_{i=1}^{52}\left(|csi_t^i| - \overline{|csi_t|}\right)^2}\sqrt{\sum\limits_{i=1}^{52}\left(|csi_{t+\tau}^i| - \overline{|csi_{t+\tau}|}\right)^2}}$$

(1)

where τ is the time period between the two CSI samples, and $|csi_t^i|$, $|csi_{t+\tau}^i|$ represent the channel gain for the i-th sub-carrier at times t and $t+\tau$, respectively. In static scenarios, for small values of τ, the similarity stays close to 1 because the CSI matches across time indicating a stable channel (Figure 2(a)). In contrast, the similarity value drops sharply under environmental or device mobility due to rapid changes in the wireless channel (Figure 2(a)). Furthermore, we found that the similarity drops faster for device mobility than environmental mobility. This happens because environmental mobility typically affects only a few multipath components, whereas if the client itself is moving, all the multipath components will be affected. Therefore, the wireless channel experiences faster variation under device mobility than under environmental mobility as evident by the larger change in CSI values.

Figure 2(b) shows the distribution of similarity of consecutive CSI samples collected at every 0.5 seconds. Clearly, the similarity distribution can be used to distinguish between static, environmental, and device-mobility scenarios. If the similarity of consecutive CSI samples is greater than $Thr_{sta} = 0.98$, we declare that the client is stationary with no environmental changes, whereas a similarity below $Thr_{env} = 0.7$ indicates device-mobility. However, from Figure 2(b), it appears that it is difficult to distinguish between device micro-mobility and macro-mobility using CSI. Figure 2(c) demonstrates that the gap between the micro and macro-mobility distribution increases if we sample the CSI more frequently. However, there are still large overlapping regions. We found that even if we sample the CSI every $50ms$, the probability of mis-classification between micro- and macro-mobility is higher than 50%. Therefore, we conclude that even if it is possible to use CSI to distinguish between device and environmental mobility, it cannot be reliably used to distinguish between different classes of device mobility.

2.4 Classifying Device Mobility using Time-of-Flight

To classify device mobility, we utilize the intuition that the distance between the client and the AP under macro-mobility changes more than that under micro-mobility. Client's distance can be estimated based on its RSSI [2] or CSI [3] measurements at the AP. However, previous work has shown that RSSI and CSI are unreliable indicators of distance [4]. In search of a better metric for distance, we find that a PHY layer metric called Time-of-Flight (ToF) can accurately capture the distance between the client and the AP [4, 5]. Due to space limitation, we provide here only a high-level overview of the
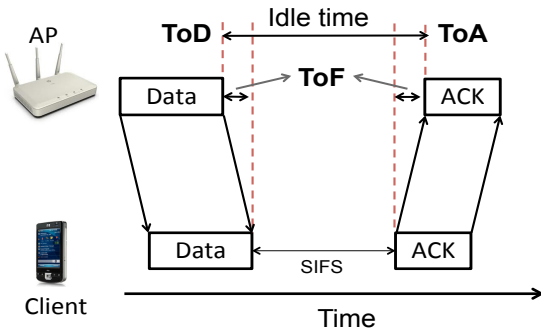
Figure 3: Computation of ToD and ToA in the PHY layer using data-ACK exchange.
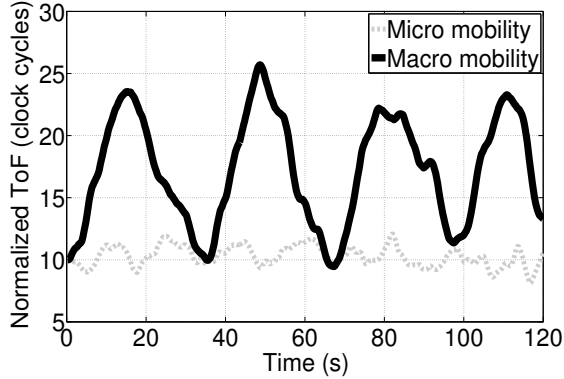

Figure 4: ToF values over time under device mobility. For the macro-mobility scenario the user walks towards and away from the AP periodically.

ToF estimation and refer the reader to [4] for details. ToF is defined as the round trip propagation time of a signal transmitted between the AP and the mobile device. The Atheros chipset can precisely compute the Time-of-Departure (ToD) of a data packet when it is sent out in the air at the PHY layer (Figure 3). On correct reception of the packet, the client waits for an SIFS duration (a fixed value in the 802.11 standard) and starts responding with an ACK packet. The chipset also reports an estimated Time-of-Arrival (ToA) of the ACK packet at the AP as shown in Figure 3. The difference between the ToA and ToD contains the ToF between the AP and the client. Since ToF is proportional to the distance between the client and the AP, it can be used to classify device-mobility.

Figure 4 plots the ToF values over time for two different device mobility scenarios. For the micro-mobility scenario, when the user naturally moves the device only within a small area, noisy ToF values can sometimes wrongly indicate changes in distance values. However, the change in noisy ToF values in the micro-mobility case is quite random, while for the macro-mobility scenario the ToF either steadily increases or decreases. This happens due to the fact that during macro-mobility a user typically walks a reasonable distance between two physical turns. Within a reasonable time interval a walking user may either approach or move away from the AP, without changing her orientation. Therefore, to detect macro-mobility, we maintain a moving window of ToF values. Only if all the ToF values in the moving window suggest an increasing or decreasing trend, we declare that the client is under macro-mobility, otherwise the client is under micro-mobility. The trend of the ToF values can also suggest the general di-
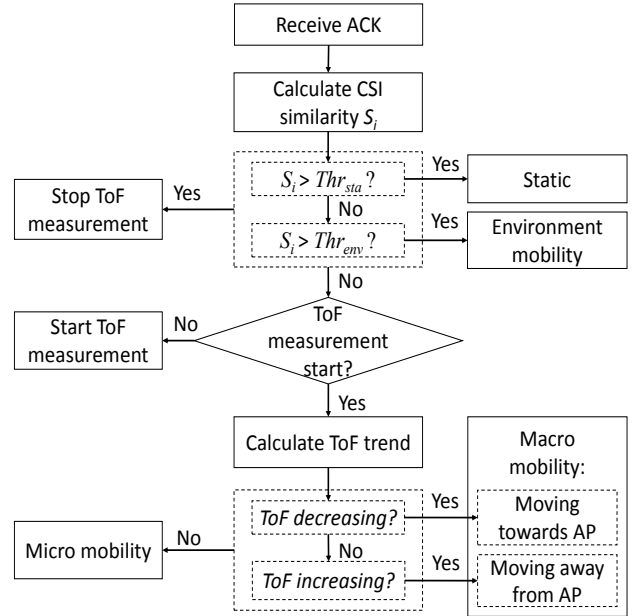

Figure 5: Mobility classification design

Ground truth	Detection result (%)			
	Static	Environmental	Micro	Macro
Static	99.7	0.3	0	0
Environmental	4.55	92.78	2.67	0
Micro-mobility	0	3.58	95.59	0.83
Macro-mobility	0	0	1.7	98.3

Table 1: Evaluation of mobility classification.

rection of the client's motion – an increasing trend means that the client is moving away from the AP and vice versa. Later, we will demonstrate how the WLAN can optimize client roaming by exploiting the user's direction of motion.

2.5 Evaluation of Mobility Classification

Figure 5 shows the overall design of mobility classification using CSI and ToF. We sample the CSI of the client periodically from existing data packet transmissions and maintain a moving average of the similarity between consecutive CSI values. We compare the CSI similarity values with empirically chosen thresholds (see Section 2.3) to distinguish between static, environmental and device mobility. If the CSI similarity indicates device mobility, we need to further consult the client's ToF values. To deal with measurement noise [4], we sample the client's ToF readings every $200ms$ and aggregate them every second using a median filter. In our implementation, if all ToF values within the past few seconds show an increasing or decreasing trend, we conclude that the client is under macro-mobility. Further, if all the ToF values show an increasing trend, we conclude that the client is moving away from the AP and vice versa.

We implemented our mobility classification algorithm using the HP MSM APs and evaluated its accuracy by performing experiments at more than 100 locations (different from those we used for learning the various thresholds and parameters of our scheme in Sections 2.3, 2.4) in two office buildings for over 24 hours. At each location we subject the client to different forms of mobility. Table 1 presents our overall performance results. We observe that the accuracy of our mobility
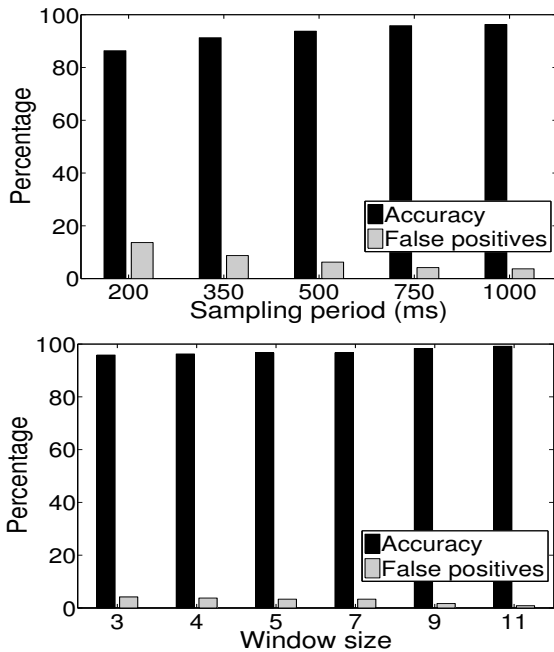
Figure 6: Accuracy and false positives of (a) CSI-based mobility detection for various CSI sampling intervals, (b) micro- and macro-mobility detection for various ToF detection window sizes.

classification is more than 92% in all scenarios.

We also found that the performance of distinguishing stationary vs. device mobility using CSI depends on the sampling period of the CSI (Figure 6(a)). The accuracy is low for short sampling period because the channel may not change very quickly even under device-mobility. We use a CSI sampling period of $500ms$ in the rest of our evaluation, yielding a median accuracy of 96%.

If the client is under device-mobility, we need to consider the ToF values within a detection window of a few seconds and determine if the ToF values have an increasing or decreasing trend. Of course, the larger the detection window, the greater will be the accuracy (Figure 6(b)), but large detection windows will delay the macro-mobility detection. Nonetheless, we find that a detection window of $4s$ yields a satisfactory accuracy of 98%, and hence we use the same in the rest of the paper.

In this work, we identify four broad categories of client mobility. One could suggest more fine-grained mobility classifications, e.g., distinguishing among different subcategories of macro-mobility based on the client's speed or among different types of micro-mobility based on whether the user is static or moves within a small area. However, we argue that in a WLAN environment categorizing client mobility in four broad categories is quite sufficient and yields simplicity in protocol design. In the following sections, we demonstrate that our broad mobility classification already offers substantial performance improvements to a number of wireless protocols.

3. MOBILITY-AWARE CLIENT ROAMING

Most wireless clients associate with the AP with the strongest RSSI value. When the RSSI falls below a predefined threshold, the client triggers a handoff, where it scans all the chan-

nels and associates with the AP with the strongest RSSI. The problem is that the client-based association control is agnostic of the user's mobility. A moving client may be close to a stronger AP, but it does not try to discover or associate with it, as long as the signal strength of the current AP is above a predefined threshold [6, 7]. Of course, if the client somehow knows that it is moving, it can scan aggressively to search for better APs [1]. But frequent scanning is time consuming and wastes energy (impacting battery life), consumes RF resources, and prevents the client from transmitting or receiving data (impacting throughput). Therefore, under mobility the client often remains wrongly connected to a far away AP, adversely affecting its own as well as the overall network performance. To address this issue, rather than designing a mobility-aware association scheme that requires changes at the client [1], we propose a controller-based protocol that moves the client to the appropriate AP whenever necessary.

3.1 Controller-based Roaming Protocol

Our roaming protocol is based on two main intuitions. First, when the client is stationary or under micro-mobility, it is less likely that a new, better AP will be discovered. Second, when the client is under macro-mobility, roaming is required only when the client moves away from the current AP. We validate these intuitions by performing more than 100 experiments at 50 different locations in our testbed. In each experiment, we subjected the client to different types of mobility and collected the RSSI values of different APs. We used the RSSI values to compute an expected throughput from different APs [8]. Figure 7(a) shows that in most mobility modes, roaming to another AP only marginally improves the performance of the client. Unless the client is walking away from the current AP, it need not roam to improve performance and can continue with the current AP. Environmental mobility can cause fluctuations of the channel between the client and the AP. However, these fluctuations are short term such that the AP with the strongest RSSI is likely to remain optimal over longer time-scales. Under micro-mobility, the client's overall physical location does not change significantly. WLAN deployments are typically well planned where each physical location is meant to be covered by a single strong AP. Therefore, roaming may be necessary only when the client's location changes significantly. But, even under macro-mobility, if the client is approaching her current AP, roaming is not necessary because the channel between the current AP and the client will only become better. On the other hand, if the client is moving away from the current AP, it may have a better channel quality to a neighboring AP. In such scenarios, the controller needs to act to move the client to the right AP.

Our controller-based protocol does not impose any changes in the client's association mechanism. The client initially connects with the strongest AP and thereafter scans if the current AP becomes weak. The current AP of the client continuously determines her mobility mode and shares it with the controller. If the current AP indicates that the client is moving away from it, the controller may have to explicitly encourage the client to roam to a better AP. To determine a candidate set of better APs, the controller instructs the neighboring APs to periodically transmit NULL data frames and compute the client's distance, RSSI and heading information towards themselves. If the client is moving towards another AP whose signal strength value is similar or higher than the current AP,
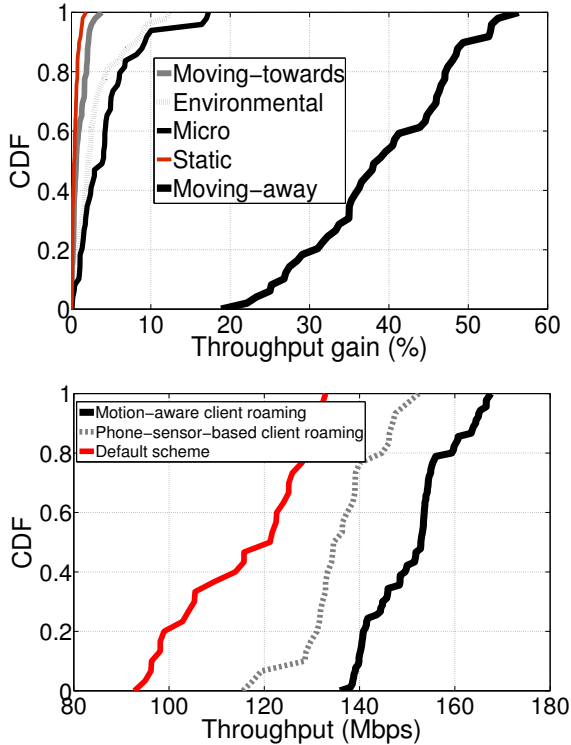
Figure 7: (a) Throughput gain by dynamically switching to the strongest AP vs. sticking to the current AP. (b) Comparison of proposed controller-based client roaming using mobility hints against existing protocols.

it is likely that the same AP will improve the client's long term performance. Note that there can be several such candidate APs which may be more appropriate (in terms of heading and signal strength) for the client. If the controller finds at least one such better AP, it instructs the current AP to disassociate the client and asks only the APs in the candidate set to respond to the client's probe request. Consequently, the client chooses an AP from the candidate set and roams to a better AP, ultimately improving performance.

3.2 Performance Evaluation

We implemented our mobility-aware client roaming protocol in our testbed of 6 HP APs (Figure 13(a)) and used a laptop connected to the APs over Ethernet as the controller. We walked naturally with the phone and injected UDP traffic from the controller to evaluate performance. We compared our scheme with the client's default roaming scheme as well as the client-based association control scheme described in [1] in which the client periodically scans for better APs if it is mobile. Figure 7(b) shows that the client's performance can be improved by assisting her to roam to a better AP when it is moving away from her current AP. Of course encouraging the client to roam requires forced disassociation from the current AP that triggers a scanning operation at the client. Scanning adds overhead. We find that the client typically completes association with a stronger AP within 200ms after being disassociated from the current AP. We believe this is reasonable because after association the client will probably use the new AP for a long duration. Nonetheless, even with the scan-

ning overhead, our proposal improves median performance by nearly 30% over the default roaming scheme.

4. MOBILITY-AWARE BIT RATE ADAPTATION

Knowledge of client's mobility state can aid bit-rate adaptation because mobility affects wireless channel conditions. The higher the intensity of motion, the faster the channel may change and the quicker the optimal bit rate becomes stale. We investigate how the optimal bit-rate of the client changes under mobility. Figure 8(a) plots the time duration for which a particular bit-rate remains optimal for different mobility modes in our testbed. For this experiment we computed the optimal rate using trace-based emulation, similar to [9]. Clearly, the optimal bit-rate changes faster if the client is mobile than when it is static. It is possible to track the optimal bit-rate in static scenarios by maintaining long success/failure history. However, under mobility, the optimal bit-rate changes quickly and hence the rate control algorithm should rely only on the recent transmission history.

We further found that depending on the client's direction of motion, i.e., moving away from or moving towards the AP, the optimal rate typically changes either upwards or downwards (Figure 8(b)). Therefore, the client's direction of motion relative to the AP can be used as a hint by the rate control algorithm. Under environmental or micro-mobility, the rate change does not show any pattern (Figure 8(c)), but the optimal bit-rate stays within a small set of values. This happens because under environmental or micro-mobility, the frequency-selective nature of the channel may change but the overall path-loss between the AP and the client remains similar. These observations suggest that current rate control schemes can improve performance by adapting according to the client's mobility mode. We demonstrate how this may be feasible by incorporating mobility-awareness in the existing Atheros MIMO rate adaptation (RA) scheme.

4.1 Atheros MIMO Rate Adaptation

Atheros MIMO Rate Adaptation (RA) is the default RA in HP MSM 460 APs. It employs a transmitter-side frame-based adaptation approach and does not require any training or client feedback. The objective of the algorithm is to determine the bit-rate that maximizes the MAC layer throughput which is calculated as bit-rate * (1-PER) where PER is the packet error rate at the given bit-rate. After each frame transmission, the algorithm updates the PER at the current bit-rate based on the number of packet failures within the aggregated frame. It applies a low pass filter on the instantaneous PER PER_{new}, as calculated from the last transmitted frame, and maintains a weighted moving average of the PER value (PER_{avg}):

$$PER_{avg} = \alpha * PER_{new} + (1 - \alpha) * PER_{avg} \qquad (2)$$

where α is called the *smoothing factor* and has a default value of $1/8$ in the Atheros RA implementation. A larger α gives a smaller weight on the previous PER history. This operation smooths out the transients and takes into account past history. The algorithm further updates the PER of other bit-rates based on the PER of the current rate. This is done by assuming that the PER is a monotonically increasing function of rate [1].

---

[1] The Atheros RA skips the MCS $5 - 7$ for single stream and MCS $8 - 10$ for double stream to maintain PER monotonicity.
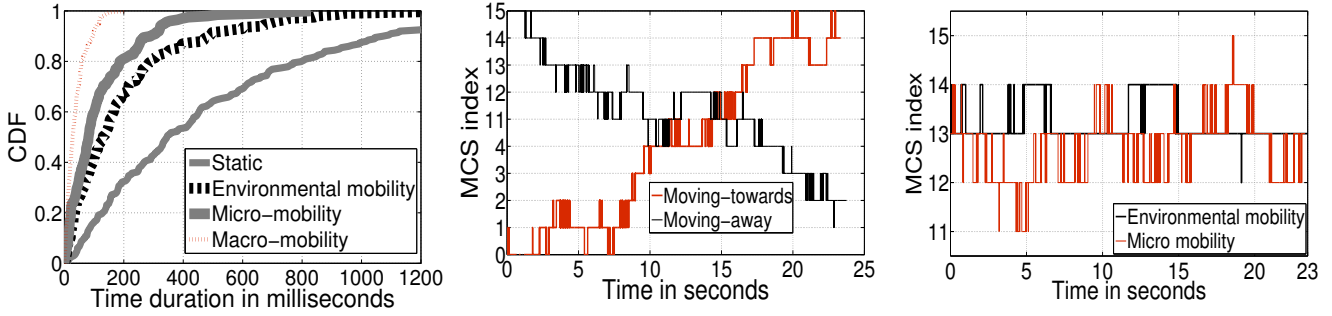
Figure 8: (a) CDF of time durations during which a current bit-rate remained optimal and no rate change was necessary. Optimal rate of a client over time when the client is under (b) macro-mobility, (c) environmental or micro-mobility.

The Atheros RA starts with the highest bit-rate and sends an aggregated frame containing several packets. If all the packets in a frame are lost such that it completely fails to get a link layer Block ACK, it often switches to the next lower rate. On the other hand, if the AP receives a Block ACK from the client, it updates the PER values and thereafter decides on a lower rate if the PER at the current rate is too high. The Atheros RA also tries to increase bit-rate whenever possible by sampling higher rates periodically. If the current rate has been successful for more than a predefined *probing interval*, the algorithm probes the next higher rate. Of course if the wireless channel is deteriorating, excessive probing may hurt performance. We show how Atheros RA can be improved using mobility hints.

4.2 Optimizing Atheros RA using Mobility Hints

Based on the observations from our measurements in the previous section (Figure 8), we make the following optimizations to the Atheros rate adaptation algorithm:

1. The Atheros RA often reduces the bit-rate if it fails to get a link layer Block ACK. This may be appropriate when the client is moving away from the AP because the wireless channel may be deteriorating. However, we note from Figures 8(b), 8(c) that if the client is moving towards the AP or if it is in any other mobility mode, this may be unnecessary because the optimal bit-rate either improves or fluctuates within a small band. In such scenarios, large errors can happen due to transient conditions such as fast fading or interference. We ignore such random effects by retrying at the current rate once or twice (Table 2) before reducing the rate, unless the client is moving away from the AP.

2. In static scenarios, we incorporate longer PER history by choosing a low value of the smoothing factor because bit-rate changes only occasionally if the client is static (Figure 8(a)). However, under mobility, the bit-rate may change quickly and thus the protocol should only account for recent packet successes and failures. Therefore, we choose a smoothing factor that is commensurate to the client's mobility mode.

3. We note that blindly probing higher rates may cause packet losses because the protocol may be already operating at the optimal rate. From Figure 8(b) we note that the optimal bit-rate of the client increases rapidly only when it is moving towards the AP. Otherwise, the optimal bit-rate seldom increases or improves only gradually. To avoid packet losses due to probing, we choose the probing interval based on the current mobility mode of the client. If the client is moving towards the AP, we use a short probing interval to quickly in-

crease the rate whenever possible. In contrast, if the client is moving away from the AP, we remain conservative by choosing a long probing interval.

4.3 Rate Adaptation Evaluation

We study the benefit of augmenting mobility hints to Atheros RA in our testbed. We modified the rate control implementation in the Atheros driver to control multiple related parameters depending on the mobility mode of the client. We conducted more than 100 experiments on 30 different links with both UDP and TCP traffic. In each experiment, we subjected the client to different types of mobility and tried different values for the Atheros RA parameters. Our final chosen parameters are shown in Table 2. We then evaluated the throughput of download TCP traffic on 150 different links in two different office buildings. At each location we subjected the client to different forms of device mobility. Figure 9(a) shows that by augmenting mobility hints to Atheros RA, it is possible to improve its median performance by 23%.

We further compare the performance of our mobility-aware rate adaptation scheme with existing state-of-the art rate control schemes. To ascertain identical channel conditions we performed trace-based emulations using techniques similar to [9]. Specifically, we modified the wireless driver of a HP laptop containing Atheros 9390 chipset to extract the CSI information from any download packet from the AP. Thereafter, we walked with the laptop in random directions for 15 minutes each and collected CSI traces for 10 such walking experiments. Using the collected traces we evaluate the throughput of Atheros RA with and without mobility hints, RapidSample [1], SoftRate [10] and ESNR [9]. Figure 9(b) shows that the mobility-aware Atheros RA performs better than RapidSample which uses sensor hints to optimize rate control. Unlike our scheme, RapidSample is unable to discriminate between fine-grained device mobility modes, e.g. micro vs. macro mobility or moving towards vs. away, and thereafter is unable to apply specific optimizations. Both SoftRate and ESNR use PHY layer information at the client to calculate the bit-rate and report it back to the AP. We find that the mobility-aware Atheros RA achieves almost the same throughput as SoftRate. ESNR outperforms SoftRate because it can indicate the bit-rate of the channel using a single observation, while SoftRate can typically only indicate whether the rate should be increased, decreased, or unchanged [11]. Our scheme can achieve 90% of ESNR's performance but notably it does not require any modifications at the client or any
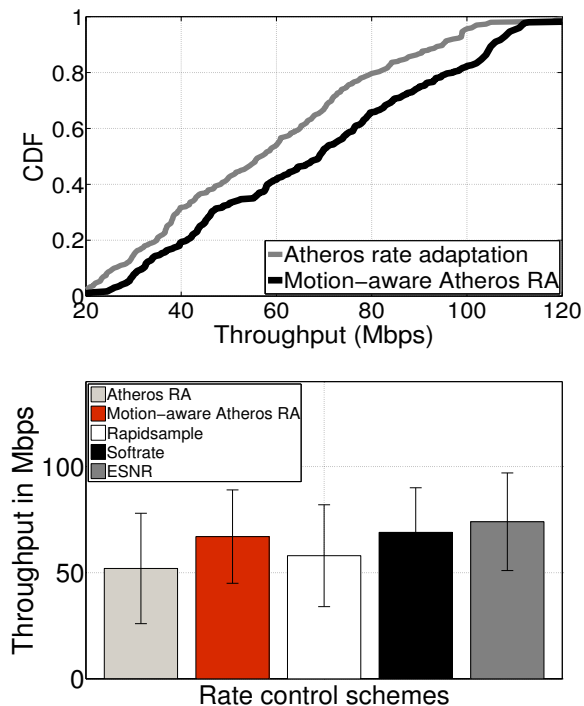
Figure 9: (a) Throughput gain by incorporating mobility-awareness in Atheros rate adaptation. (b) Comparison of mobility-aware Atheros RA against other state-of-the art schemes.

per-client calibration to establish the ESNR vs. bit-rate relationship [9].

5. MOBILITY-AWARE FRAME AGGREGATION

To achieve high throughput, IEEE 802.11 allows the wireless transmitter to aggregate multiple packets in a single frame. Aggregation achieves higher throughput because the PHY protocol overhead over multiple packets can be amortized by packing them into a single frame. As the number of packets in a frame (often referred to as aggregation size) increases, the efficiency of the protocol improves resulting in greater performance. Thus, it is generally believed that choosing a larger aggregation size is beneficial. In fact, current implementations allow the transmitter to aggregate as many packets as it can within an aggregation time of 10ms [12]. However, choosing the highest aggregation size may hurt performance under mobility as the wireless channel may vary within an aggregated frame.

To understand the effect of channel variations on aggregation, we conducted a total of 60 experiments on 20 different links in our office building. We injected download traffic to the client while varying the aggregation size defined in the driver from $2ms$ to $8ms$ for each mobility mode. We found that in mobility scenarios, performance can actually decrease if a large aggregation size is used (Figure 10(a)). This happens because the channel equalization procedure employed by today's wireless radios estimates the channel only at the beginning of the frame. Under mobility, it is quite possible that the wireless channel changes by the end of frame resulting into packet losses.
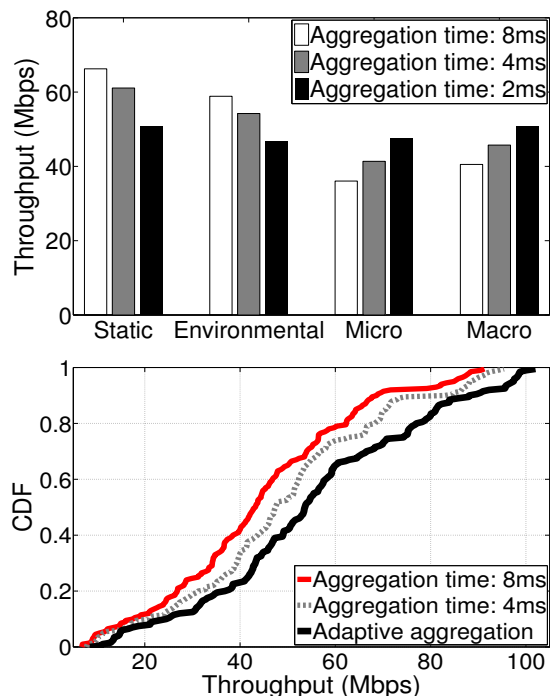

Figure 10: (a) Mean throughput vs. frame aggregation time in different mobility scenarios. (b) Comparison of the adaptive mobility-aware frame aggregation scheme against schemes that use a statically configured aggregation time.

5.1 Optimization and Performance Evaluation

To optimize the performance of frame aggregation, we note from Figure 10(a) that the optimal aggregation time varies across different classes of mobility. The higher the intensity of mobility, the lower is the channel coherence time and hence the lower is the optimal aggregation time. Therefore, we propose to tune the aggregation size at the AP based on the mobility state of the client. We modified the WiFi driver at the AP to vary the maximum allowed aggregation time based on the mobility mode of the client – 8ms for static and environmental mobility and 2ms for micro- and macro-mobility. The actual aggregation size depends on the bit-rate used to communicate to the client: Aggregation size = Maximum allowed aggregation time / Bit-rate.

We evaluated our modifications on 50 different links in two office buildings and subjected the client to various mobility modes, while injecting download TCP traffic from the AP. Figure 10(b) plots the performance of our aggregation scheme on 50 different links. By selecting the aggregation size based on mobility, it is possible to improve the median throughput performance by 15% over the default Atheros scheme which uses a statically configured $4ms$ maximum aggregation time.

6. MOBILITY-AWARE BEAMFORMING AND MU-MIMO

MIMO beamforming can improve the performance of a wireless link by exploiting multiple antennas at the transmitter. Single-user (SU) beamforming in 802.11n precodes a single packet across multiple antennas at the AP so that the signals
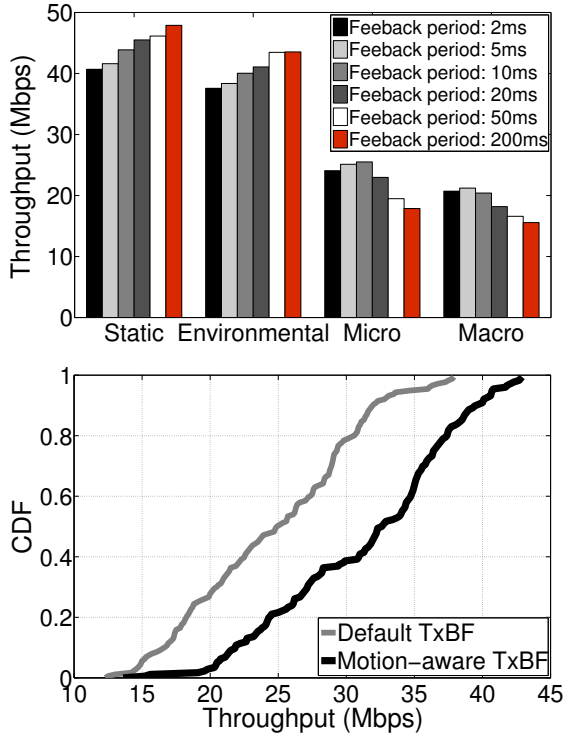
Figure 11: (a) Average throughput vs. CSI feedback period in different mobility scenarios using transmit beamforming. (b) CDF of throughput gain for 300 different links subjected to a variety of mobility modes.
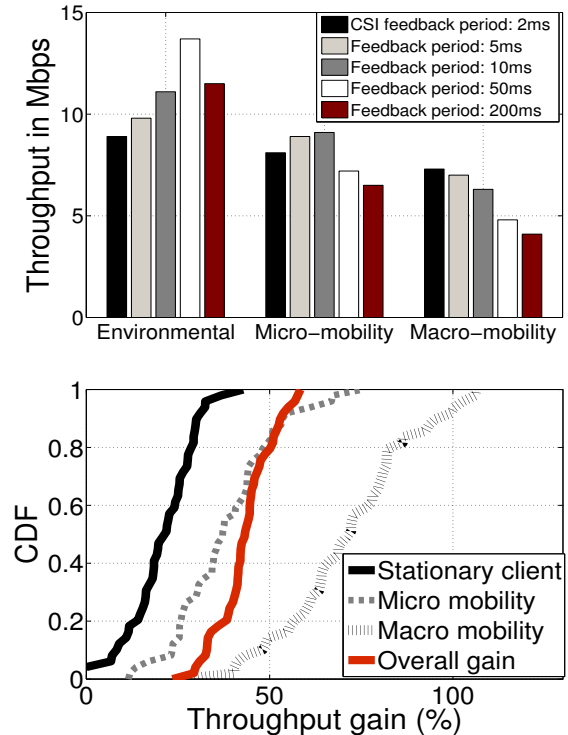
Figure 12: (a) Average throughput vs. CSI feedback period using MU-MIMO techniques to serve three different clients. (b) CDF of throughput gain for 200 emulation scenarios. Each scenario includes 3 clients, one each in environmental, micro-, and macro-mobility mode.

from different antennas combine coherently at the client, enhancing performance. Similarly, 802.11ac multi-user MIMO (MU-MIMO) precodes packets for different clients so that they can be transmitted simultaneously. The precoding operation combines the outgoing packets and carefully chooses the weights based on the CSI values between the AP (transmitter) and all the clients (receivers). Therefore, for effective beamforming, the AP must obtain timely CSI feedback from all the clients. The CSI feedback packet may consist of a real and imaginary value (quantized into up to 8 bits) for each subcarrier and transmit-receive antenna pair. Frequent CSI feedback may be harmful because the feedback packet is typically transmitted at the lowest bit-rate, consuming significant channel airtime. On the other hand, infrequent feedback may often result in stale CSI and consequently performance loss. Hence managing the CSI feedback period is critical to the performance of MIMO beamforming.

6.1 Effect of Mobility on SU-Beamforming

To understand the effect of CSI feedback period on SU-Beamforming, we conducted 150 experiments on more than 50 different links in our office building by allowing the AP to periodically initiate CSI feedback from the client and employ beamforming while transmitting the data packets. We found that none of the popular smartphones support explicit beamforming and hence we used an AP as the client in our beamforming experiments. Figure 11(a) shows that in static scenarios, a shorter CSI feedback period can actually hurt performance. This is because the wireless channel remains stable in static scenarios. Therefore, frequent feedback only adds

to the overhead and does not always result in better CSI estimates. In contrast, if the client is mobile, the wireless channel can change quickly, requiring frequent CSI feedback. Thus, we need to adapt the CSI feedback period depending on the client's mobility mode.

6.2 Effect of Mobility on MU-MIMO

Our AP platform only supports 802.11a/b/g/n and hence is not amenable to MU-MIMO experiments. We resort to trace-based emulations to understand the effect of feedback period on MU-MIMO. We emulate a scenario where the AP has 3 antennas and employs MU-MIMO techniques to concurrently serve 3 clients – one each in environmental, micro- and macro-mobility scenarios. For the emulation, we simultaneously collected CSI traces at 3 locations using laptops containing Atheros 9390 chipset and a single receive antenna. We instrumented the Atheros driver used in the laptop to report the CSI values for every transmitter-receiver antenna pair for each data packet received from the AP. Thereafter we sent data packets from the AP to each laptop every 500us. We fed the series of CSI values to a MU-MIMO emulator written in C. The emulator simulates CBR traffic between the AP and the clients and reports the throughput for each client with and without MU-MIMO techniques.

Figure 12(a) shows that MU-MIMO is also sensitive to stale CSI values. Interestingly, mobility only affects the performance of the mobile client and does not impact the static clients noticeably. This happens because MU-MIMO precoding ensures that signals from all the transmit antennas com-

bine coherently at each intended client, which is only re
lated to the channel between the AP and the respective client
Therefore it is possible to improve MU-MIMO performance by
using different CSI feedback periods that are commensurate
to the mobility modes of individual clients.

6.3 Performance Optimization & Evaluation

To optimize performance using SU-beamforming and MU-MIMO
we note from figures 11(a) and 12(a) that the optimal CSI
feedback period varies across different classes of mobility. The
higher the intensity of mobility, the quicker the CSI becomes
stale and hence the higher is the required frequency of the CSI
feedback. Therefore, we propose to tune the CSI feedback pe
riod at the AP based on the mobility state of the client. We
modified the WiFi driver at the AP to vary the feedback period
based on the mobility mode of the client – 200ms for static,
50ms for environmental mobility, 10ms for micro-, and 5ms
for macro-mobility.

We evaluated SU-beamforming using adaptive CSI feedback
on 300 different mobile links in two office buildings by inject-
ing download TCP traffic from the AP. Figure 11(b) shows
that by selecting the feedback period based on mobility, it
is possible to improve the median throughput by 33% over
the default Atheros scheme which uses a statically configured
$200ms$ feedback period.

To evaluate the benefits of adaptive CSI feedback on MU-
MIMO, we used the trace based emulation setup described
in section 6.2. In total, we collected CSI traces from 50 uni-
formly chosen locations from a busy office environment. At
each location, we kept the client stationary for approximately
5 minutes, followed by micro-mobility and macro-mobility ex-
periments for the same interval. Apart from collecting the CSI
trace at the client, we also recorded the CSI and ToF values
from the data-ACK exchange at the AP. For each trace, we es-
timate the mobility mode of the client every second using our
techniques described in section 2. Thereafter we randomly
pick 3 client locations and feed the corresponding CSI traces
and mobility estimates to our MU-MIMO emulator. To cal-
culate the precoding vector, the MU-MIMO emulator samples
the CSI traces every feedback period. For our scheme, the
feedback period is chosen on a per client basis depending
on her mobility state – 200ms for static, 50ms for environ-
mental mobility, 10ms for micro-, and 2ms for macro-mobility,
whereas the default scheme uses a 200ms CSI feedback period
for all clients. The emulator uses Atheros RA for rate control
and does not employ aggregation. We repeated the emula-
tion 200 times, each time with a different set of 3 client lo-
cations. Figure 12(b) plots the throughput gain of MU-MIMO
using adaptive CSI feedback over the static feedback period
scheme. Performance gain is evident for all varieties of mobil-
ity, with the most gains for clients under macro-mobility. On
average, our scheme improves MU-MIMO's network through-
put by close to 40%.

7. OVERALL PROTOCOL PERFORMANCE

As explained in the preceding sections, we implement the
client's mobility mode detection scheme at the AP. Further,
depending on the client's mobility mode, at the AP and the
WLAN controller, we implement several actions to improve
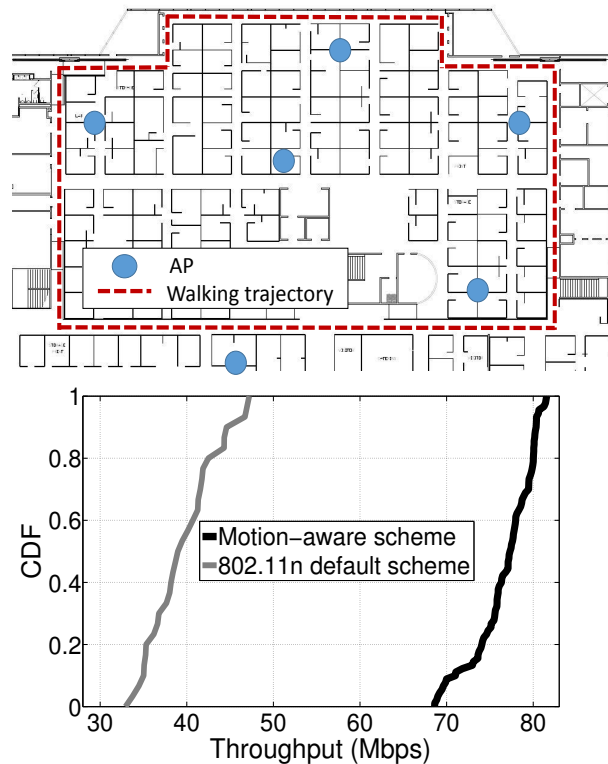rate control, client roaming, aggregation and beamforming

**Figure 13: (a) Testbed setting for overall performance
evaluation. (b) CDF of end-to-end throughput for** 90 **tests
with and without mobility hints.**

(Table 2)[2]. To evaluate the combined protocol performance
for all such optimizations, we conducted iperf-based system
throughput tests in our testbed using 6 APs (Figure 13(a))
with UDP traffic. We walked naturally with the client through
different parts of the trajectory shown in Figure 13(a) and
evaluated the performance of our system as well as the de-
fault system which does not distinguish client's mobility. As
shown in Figure 13(b), the mobility-aware scheme outper-
formed the default system in all the 90 tests we conducted
in our testbed. By taking appropriate actions based on the
client's mobility mode, we improve overall protocol perfor-
mance by almost 100%.

8. RELATED WORK

Sensor hint-based optimization: The work closest to ours
is [1], which uses sensor hints (e.g., from GPS, accelerom-
eters, gyroscopes) from mobile devices to augment wireless
protocols. For example, the authors use mobility hints from
the accelerometer to select between two rate adaptation al-
gorithms; SampleRate [13] (in static environments) or Rapid-
Sample (in mobile environments). They further use accelerom-
eter hints to implement a hint-aware roaming protocol. Our
results in sections 3 and 4 demonstrate that due to our PHY-
based fine-grained discrimination of mobility modes, we out-
perform both RapidSample and the client-based roaming pro-
tocol from [1]. Further, while [1] only considers legacy
802.11, we show how mobility hints can help high-throughput

---

[2]We could not implement our MU-MIMO optimizations in the
testbed because of lack of 802.11ac support.

Protocol	Parameter	Mobility type				
		Static	Environment	Micro mobility	Macro mobility	
					Moving away	Moving towards
Client roaming	Roaming preparation	no	no	no	Encourage client to roam to better AP if available.	no
Rate adaptation	Probe interval	50ms	50ms	50ms	100ms	20ms
	PER smoothing factor	1/16	1/12	1/4	1/3	1/3
	Rate retries	1	1	1	0	2
Frame aggregation	Aggregation limit	8ms	8ms	2ms	2ms	2ms
Beamforming	CV update interval	200ms	50ms	10ms	5ms	5ms
MU-MIMO	CV update interval	200ms	50ms	10ms	2ms	2ms

Table 2: Summary of mobility-aware protocol actions.

mechanisms introduced in the 802.11n/ac standards, such as frame aggregation, beamforming and MU-MIMO. Trinity [14] also uses accelerometer-based mobility hints to optimize distributed antenna systems and network-MIMO. Mobisteer [15] uses location hints from GPS to find the best directional antenna orientation and the AP to associate with. Breadcrumbs [16] builds a mobility model based on GPS coordinates and uses it to predict the best AP to associate with. CARS [17] uses GPS readings to optimize rate control. In contrast to existing works, we exploit PHY layer information available from off-the-shelf APs to obtain the client's mobility information, without relying on any external sensor and without requiring any coordination with the client.

PHY-based mobility detection: Recently, there has been a lot of interest in developing device-free passive motion detection systems using radio signal dynamics. While most proposals require hardware modifications [18–20], [21–24] use RSSI, and FIMD [25] proposes CSI-based techniques. These systems can only identify environmental mobility and often have higher complexity than our system. For example, FIMD uses feature extraction and data fusion from multiple links to detect environmental mobility with 90% accuracy. We can classify among static, environmental, and device mobility by only calculating the similarity of the CSI values from a single link.

Rate control and client roaming: A large number of RA algorithms have been proposed in the past. They can be classified into frame-based [13, 26–31], SNR-based [9, 32–34], and PHY-layer based [9–11]. Frame-based algorithms are often used in practice due to their simplicity but they are slow to respond to rapidly changing channel conditions in mobile environments. SNR-based algorithms can adapt faster but they need "in-situ" training to overcome this sensitivity [35]. PHY-layer based approaches have been shown to outperform SNR-based algorithms but they cannot be implemented on commodity APs. ESNR [9] uses CSI from commodity cards to estimate the bit-rate, but it requires careful per-client calibration to obtain the ESNR vs. bit-rate relationship. Our mobility-aware modifications of the frame-based Atheros RA algorithm does not require any such calibration and achieves 90% of ESNR's performance on average. Like rate control, client roaming has been widely studied to improve wireless performance [36–39]. However, unlike our scheme, most proposals either require changes at the client [37, 38] or at both the AP and the client [39]. Like DenseAP [40], we manage the client's association from the controller with no modifications at the client. We further improve controller-based association control by incorporating mobility hints.

802.11n/ac enhancements: Impact of channel dynamics on 802.11n frame aggregation have been studied using simulation [41], modeling [42, 43] and testbed experiments [30]. [42] and [43] calculate the frame aggregation limit based on the bit error-rate (BER) and SNR respectively. While BER and SNR can be indirect indicators, we observe that mobility-hints can be a direct indicator of aggregation limit, improving wireless performance. CSI feedback remains a critical issue for high throughput 802.11n/ac mechanisms such as beamforming and MU-MIMO [44]. While there has been work towards experimental evaluation of MU-MIMO [45] and its variants [46], decreasing the overhead of CSI feedback is mostly limited to [44]. While [44] is limited to frequent CSI sampling, we show that by augmenting CSI with ToF, mobility hints can reduce the overhead due to the CSI feedback period.

9. DISCUSSION AND LIMITATIONS

Uplink traffic: In this work we focused on downlink traffic because download is still the dominant type of traffic in today's WLANs – often 20 times more than uplink traffic [47]. While some of our mobility-aware protocol modifications are mostly limited to the AP (e.g., controller-assisted roaming, beamforming, MU-MIMO), the others, such as bit-rate adaptation and frame aggregation can also be implemented on the client side as well to benefit uplink traffic.

Roaming for real time traffic: Our roaming protocol explicitly encourages the client to roam whenever it detects that the client is moving towards a better AP. While this approach improves throughput for bulk transfer applications, as we discussed in section 3, the 200 ms required for a handoff may increase both jitter and packet loss for real-time traffic. One solution to this problem will be to minimize the number of handoffs by choosing the farthest, yet reasonably strong, AP the client is moving towards. Alternatively, we note that fast BSS transition in 802.11r reduces the handoff overhead to only 40ms [48]; hence, clients such as Apple devices that support 802.11r [49] can immediately benefit from our roaming protocol without causing disruptions to real-time traffic.

Moving on a circle around the AP: If a client is moving on a circle around the AP, our system will wrongly classify the type of mobility as micro- instead of macro-mobility, as the ToF values will not be characterized by an increasing or decreasing trend. Although this is not a typical mobility pattern, in our future work, we plan to augment our system with Angle of Arrival (AoA) information [3, 50] to address this limitation.

Other benefits of mobility-awareness: Mobility-awareness could also guide the selection of channel width (a narrow 20 MHz channel may be more robust than the wider 40MHz,

80MHz, or 160 MHz channels, and hence, it may be preferred when the client is moving away from the AP) and the type of MIMO mode (spatial diversity may be preferred over spatial multiplexing when the client is moving away from the AP). However, our preliminary experiments did not show any significant gains for these two cases. We plan to investigate these in more detail in our future work. We also plan to study other protocols that could benefit from mobility-awareness, e.g., adapting the length of cyclic prefix based on the client speed, scheduling client traffic at an AP taking movement into account, network monitoring, as well as distributed MIMO techniques.

10. CONCLUSION

Today's WLANs have to be able to distinguish different types of client mobility and use adaptive, motion-aware protocols in order to sustain high performance. In this paper, we demonstrate how different client mobility modes can be distinguished by using physical layer information – Channel State Information (CSI) and Time-of-Flight (ToF) – available at commodity APs, with no modifications on the client side. Our testbed experiments show that our algorithm achieves more than 90% accuracy in a variety of scenarios. We also demonstrate how fine-grained motion determination can improve performance of client roaming, bit-rate control, frame aggregation, and MIMO beamforming in 802.11n/ac WLANs. Our extensive evaluation shows that motion-aware AP-side protocols outperform their motion-oblivious counterparts as well as prior mobility-aware client-side solutions. In particular, the combination of all four mobility-aware protocols improves throughput by almost 100% over the default AP implementation using motion-oblivious protocols.

11. ACKNOWLEDGMENTS

12. REFERENCES

[1] Hari Balakrishnan, Lenin Ravindranath, Calvin Newport, and Samuel Madden. Improving Wireless Network Performance Using Sensor Hints. In *Proc. of USENIX NSDI*, 2011.

[2] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *Proc. of ACM MobiCom*, 2010.

[3] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding WiFi localization. In *Proc. of ACM MobiSys*, 2013.

[4] Alex Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. Single access point based indoor localization. In *Proc. of ACM MobiSys*, 2014.

[5] Domenico Giustiniano and Stefan Mangold. Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless LAN. In *Proc. of ACM CoNEXT*, 2011.

[6] Ishwar Ramani and Stefan Savage. Syncscan: practical fast handoff for 802.11 infrastructure networks. In *Proc. of IEEE INFOCOM*, 2005.

[7] Minho Shin, Arunesh Mishra, and William A. Arbaugh. Improving the latency of 802.11 hand-offs using neighbor graphs. In *Proc. of ACM MobiSys*, 2004.

[8] Souvik Sen, Bozidar Radunovic, Jeongkeun Lee, and Kyu-Han Kim. Cspy: finding the best quality channel without probing. In *Proc. of ACM MobiCom*, 2013.

[9] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *Proc. of ACM SIGCOMM*, 2010.

[10] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. Cross-Layer Wireless Bit Rate Adaptation. In *Proc. of ACM SIGCOMM*, 2009.

[11] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. AccuRate: Constellation Based Rate Estimation in Wireless Networks. In *Proc. of USENIX NSDI*, 2010.

[12] Yang Xiao. IEEE 802.11n: enhancements for higher throughput in wireless LANs. *IEEE Wireless Communications*, 12(6):82–91, 2005.

[13] John C. Bicket. Bit-rate selection in wireless networks. Master's thesis, Massachusetts Institute of Technology, February 2005.

[14] Shailendra Singh, Karthikeyan Sundaresan, Amir Khojastepour, Sampath Rangarajan, and Srikanth V. Krishnamurthy. One strategy does not serve all: tailoring wireless transmission strategies to user profiles. In *Proc. of ACM HotNets*, 2012.

[15] Vishnu Navda, Anand Prabhu Subramanian, Kannan Dhanasekaran, Andreas Timm-Giel, and Samir R. Das. MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access. In *Proc. of ACM MobiSys*, 2007.

[16] Anthony J. Nicholson and Brian D. Noble. BreadCrumbs: Forecasting Mobile Connectivity. In *Proc. of ACM MobiCom*, 2008.

[17] Pravin Shankar, Tamer Nadeem, Justinian Rosca, and Liviu Iftode. CARS: Context-Aware Rate Selection for Vehicular Networks. In *Proc. of IEEE ICNP*, 2008.

[18] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proc. of ACM MobiCom*, 2013.

[19] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. Bringing gesture recognition to all devices. In *Proc. of USENIX NSDI*, 2014.

[20] Fadel Adib and Dina Katabi. See through walls with WiFi!. In *Proc. of ACM SIGCOMM*, 2013.

[21] Michael Wallbaum and Stefan Diepolder. A Motion Detection Scheme For Wireless LAN Stations. In *Proc. of IEEE ICMU*, 2006.

[22] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. Challenges: Device-free Passive Localization for Wireless Environments. In *Proc. of ACM MobiCom*, 2007.

[23] Jie Yang, Yong Ge, Hui Xiong, Yingying Chen, and Hongbo Liu. Performing Joint Learning for Passive Intrusion Detection in Pervasive Wireless Environments. In *Proc. of IEEE INFOCOM*, 2010.

[24] Ahmed E. Kosba, Ahmed Saeed, and Moustafa Youssef. RASID: A Robust WLAN Device-free Passive Motion Detection System. In *Proc. of IEEE PerCom*, 2012.

[25] Jiang Xiao, Kaishun Wu, Youwen Yi, Lu Wang, and Lionel M. Ni. FIMD: Fine-grained Device-free Motion Detection. In *Proc. of IEEE ICPADS*, 2012.

[26] Ad Kamerman and Leo Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, 2(3):118–133, 1997.

[27] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. of ACM MobiCom*, 2001.

[28] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and Daji Qiao. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *Proc. of IEEE INFOCOM*, 2006.

[29] Starky H.W. Wong, Hao Yang, Songwu Lu, and Vaduvur Barghavan. Robust rate adaptation for 802.11 wireless networks. In *Proc. of ACM MobiCom*, 2006.

[30] Ioannis Pefkianakis, Yun Hu, Starsky H.Y. Wong, Hao Yang, and Songwu Lu. MIMO Rate Adaptation in 802.11n Wireless Networks. In *Proc. of ACM MobiCom*, 2010.

[31] Duy Nguyen and J.J. Garcia-Luna-Aceves. A practical approach to rate adaptation for multi-antenna systems. In *Proc. of IEEE ICNP*, 2011.

[32] Jiansong Zhang, Kun Tan, Jun Zhao, Haitao Wu, and Yongguang Zhang. A Practical SNR-Guided Rate Adaptation. In *Proc. of IEEE INFOCOM*, 2008.

[33] Glenn Judd, Xiaohui Wang, and Peter Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *Proc. of ACM MobiSys*, 2008.

[34] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. Joint Rate and Channel Width Adaptation for 802.11 MIMO Wireless Networks. In *Proc. of IEEE SECON*, 2013.

[35] Joseph Camp and Edward Knightly. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-layer Implementation and Experimental Evaluation. In *Proc. of ACM MobiCom*, 2008.

[36] Minkyong Kim, Zhen Liu, Srinivasan Parthasarathy, Dimitrios Pendarakis, and Hao Yang. Association control in mobile wireless networks. In *Proc. of IEEE INFOCOM*, 2008

[37] Sudarshan Vasudevan, Konstantina Papagiannaki, Christophe Diot, Jim Kurose, and Don Towsley. Facilitating access point selection in IEEE 802.11 wireless networks. In *Proc. of ACM IMC*, 2005.

[38] George Athanasiou, Thanasis Korakis, Özgür Erçetin, and Leandros Tassiulas. Dynamic cross-layer association in 802.11-based mesh networks. In *Proc. of IEEE INFOCOM*, 2007.

[39] Yigal Bejerano, Seung-Jae Han, and Li Erran Li. Fairness and load balancing in wireless LANs using association control. In *Proc. of ACM MobiCom*, 2004.

[40] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, and Brian Zill. Designing high performance enterprise Wi-Fi networks. In *Proc. of USENIX NSDI*, 2008.

[41] Boris Ginzburg and Alex Kesselman. Performance analysis of A-MPDU and A-MSDU aggregation in IEEE 802.11n. In *Proc. of IEEE Sarnoff Symposium*, 2007.

[42] Xin He, Frank Y Li, and Jiaru Lin. Link adaptation with combined optimal frame size and rate selection in error-prone 802.11n networks. In *Proc. of IEEE ISWCS*, 2008.

[43] Feng Kai-Ten, Lin Po-Tai, and Liu Wen-Jiunn. Frame-aggregated link adaptation protocol for next generation wireless local area networks. *EURASIP Journal on Wireless Communications and Networking*, 2010:10:1–10:13, 2010.

[44] Xiufeng Xie, Xinyu Zhang, and Karthikeyan Sundaresan. Adaptive feedback compression for MIMO networks. In *Proc. of ACM MobiCom*, 2013.

[45] Hariharan Shankar Rahul, Swarun Kumar, and Dina Katabi. JMB: scaling wireless capacity with user demands. In *Proc. of the ACM SIGCOMM*, 2012.

[46] Horia Vlad Balan, Ryan Rogalin, Antonios Michaloliakos, Konstantinos Psounis, and Giuseppe Caire. Achieving high data rates in a distributed MIMO system. In *Proc. of ACM MobiCom*, 2012

[47] Ning Ding, Daniel Wagner, Xiaomeng Chen, Y. Charlie Hu, and Andrew Rice. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In *Proc. of ACM SIGMETRICS*, 2013.

[48] Sangeetha Bangolae, Carol Bell, and Emily Qi. Performance study of fast BSS transition using IEEE 802.11r. In *Proc. of ACM MobiCom*, 2006.

[49] Deploying iPhone and iPad Wi-Fi. In *https://www.apple.com/kr/iphone/business/docs/ ios_6_wifi_sep12.pdf*.

[50] Jie Xiong and Kyle Jamieson. Arraytrack: a fine-grained indoor location system. In *Proc. of USENIX NSDI*, 2013.