2014 International Conference on Computing, Networking and Communications, Wireless Networks Symposium



# FastBeam: Practical Fast Beamforming for Indoor Environments

Chao-Fang Shih and Raghupathy Sivakumar
Georgia Institute of Technology, Atlanta, GA, U.S.A

***Abstract*—In this work, we explore algorithms to perform fast beamforming when relying only on RSSI measurements in indoor environments. Fast beamforming allows for a faster adaptation of the antenna weights for environments where the channel is time varying. Time varying channel can be caused by client mobility, orientation changes at the client, and environmental changes. We show that existing techniques for RSSI based beamforming have a high time complexity, and hence severely under-perform when the channel is time varying. We then propose FastBeam, a suite of strategies that in tandem enable fast and practical beamforming using only RSSI measurements. FastBeam consists of both optimality preserving techniques and heuristic approaches that are selectively applied depending on the rate of change of the channel. We implement FastBeam on a Phocus Array System with eight antennas and use experimental evaluations to study its performance.**

## I. Introduction

Transmit beamforming enables a transmitter to direct signals at an intended receiver; thus, using beamforming can improve the signal quality and hence the data throughput and range for a wireless link [1]. Even in indoor environments, where directional antennas with fixed radiation patterns fail to provide performance improvements due to multipath fading, beamforming can still deliver substantial benefits through the adjustment of the relative weights and phases of the different elements of an antenna array. By default, beamforming requires the complete channel state information (CSI). However, collecting such information at the receiver requires specialized hardware and complex processing. Recent techniques [2] have demonstrated that optimal beamforming can be achieved while relying purely on the receive signal strength indicator (RSSI), a parameter that is readily and easily available on most wireless cards.

In this work, we first show that existing techniques for RSSI based beamforming do not reach achievable performance in *time varying channels*, and can in fact perform worse than omni-directional antennas under some scenarios. We then explore algorithms to perform *fast beamforming* when relying only on RSSI measurements. Fast beamforming allows for a faster continuous adaptation of the antenna weights for environments where the *channel is time varying* because of client mobility, client orientation changes, and environmental changes (such as changes in positions of objects/people).

We thus present FastBeam, a suite of strategies that in tandem enables practical beamforming in time varying environments. FastBeam uses only RSSI measurements for performing beamforming and consists of both optimality preserving techniques and heuristic approaches that are selectively applied depending on the rate of change of the channel. We implement $FastBeam$ on a Phocus Array System ([3]) with eight antennas and use experimental evaluations to study its performance in indoor environments. We show that the time complexity for RSSI based beamforming can be reduced by an average of $50\%$ and by as much as $75\%$ compared to existing approaches ([2]). Such a reduction in the time complexity has a significant impact on the throughput performance under several conditions: we show that the throughput performance of $FastBeam$ compared to the existing work is $1.4\times$ better on average, and up to $1.8\times$ better in the best case.

**Related Work**: Related works consider the usage of smart antenna technologies for indoor wireless LANs. However, they either use directional antennas with fixed radiation patterns ([4], [5], and [6]) that are less sophisticated than beamforming, or require the usage of CSI ([7]) that require the clients to be equipped with special hardware. In [2], an algorithm is presented to enable an off-the-shelf client to gain beamforming benefits without hardware modifications. However, we show that the proposed algorithm (called $SimpleBeam$ in this paper) is not suitable for time varying channels.

## II. Background of RSSI Based Beamforming

### A. Beamforming with RSSI

Beamforming techniques can deliver considerable improvements over omni-directional (omni) communication in both outdoor and indoor environments. Specifically, in indoor environments, while simple directional antenna solutions suffer from multipath fading, adaptive beamforming still holds tremendous promise. A recent advancement in beamforming is a partial CSI approach called *RSSI based beamforming* [2], which only uses RSSI measurements sent as feedback from the client to make beamforming decisions. While it is provably optimal (same performance as traditional beamforming under static channel conditions), the most attractive property of RSSI based beamforming is that it does not require the receiver to be equipped with an antenna array or special hardware to provide benefits. This makes the technique attractive to cater to off-the-shelf wireless clients.

### B. Algorithm for RSSI based Beamforming

Assuming that there are $n$ antenna elements at the transmitter, and a single antenna at the receiver, the consequent Mul-

This work was funded in part by the National Science Foundation under grant CNS-1017984.

tiple Input Single Output(MISO) channel can be represented as follows:

$$y = \mathbf{hw}s + z, \tag{1}$$

where $y$ is the received signal, $z$ is the additive White Gaussian noise, the $1 \times n$ vector of complex number $\mathbf{h} = [h_0, h_1, \cdots h_{n-1}]$ is the respective channel gains between each transmitter antenna and the receiver antenna, and the $n \times 1$ vector of complex numbers $\mathbf{w} = [w_0, w_1, \cdots w_{n-1}]^T$ is a beamformer that translates the transmit symbol $s$ to the transmitted signals $\mathbf{x} = \mathbf{w}s$. The optimal setting of beamformer $\mathbf{w}$ is to let each beamformer weight $w_i$ become the complex conjugate of $h_i$, so that the signals from each channel combine coherently and reinforce each other at the receiver.

RSSI based beamforming does channel estimation and calculates the beamformer $\mathbf{w}$ according to the RSSI values that are sent as feedback from Rx. The core idea is to use both single and tandem-activated antennas to determine the channel magnitude and phase difference between the antennas. Since RSSI is a good approximation of the received power, we could estimate the received power according to the RSSI value. When a single antenna is activated, the respective received powers $P_i = |h_i|^2$ (assuming the Tx power is unity) are obtained from the RSSI value sent as feedback from Rx. When antennas $i$ and $j$ are tandem activated (with antenna phase difference $\rho_{ij}$ set to 0), the received powers $P_{ij} = |h_i + h_j|^2$ are obtained. $P_{ij}$ can be written as:

$$P_{ij} = P_i + P_j + 2\sqrt{P_i P_j}\cos(\theta_{ij}), \tag{2}$$

where $\theta_{ij}$ is the channel phase difference between $h_i$ and $h_j$. By rewriting equation (2), we can get $\theta_{ij}$ using $P_i$, $P_j$, and $P_{ij}$:

$$\cos\theta_{ij} = (\frac{P_{ij} - P_i - P_j}{2\sqrt{P_i P_j}}), \tag{3}$$

$$\phi_{ij} = \cos^{-1}(\frac{P_{ij} - P_i - P_j}{2\sqrt{P_i P_j}}). \tag{4}$$

Since $|\cos\theta| = |\cos(-\theta)| = |\cos(\pi-\theta)| = |\cos(-\pi+\theta)|$, there are 4 possible cases: $\theta_{ij} = \phi_{ij}$, $\theta_{ij} = -\phi_{ij}$, $\theta_{ij} = \pi - \phi_{ij}$, or $\theta_{ij} = -\pi + \phi_{ij}$. In $SimpleBeam$ ([2]), this ambiguity is solved through another 4 sets of tandem activated antennas. Let $P_{ij\_\rho_{ij}}$ represent the received power of tandem activated antennas $i$ and $j$ with antenna phase difference set to $\rho_{ij}$ ($P_{ij\_\rho_{ij}} = P_i + P_j + 2\sqrt{P_i P_j}\cos(\theta_{ij} + \rho_{ij})$). $P_{ij\_\phi_{ij}}$, $P_{ij\_-\phi_{ij}}$, $P_{ij\_\pi-\phi_{ij}}$, and $P_{ij\_-\pi+\phi_{ij}}$ are measured, and the antenna phase setting that corresponds to the maximum received power indicates the value of $-\theta_{ij}$. (Actually, since the exact value of $\cos\theta$ can be known, 2 sets of tandem activated antennas is sufficient. However, since time-complexity is not a concern in [2], it does not deal with this issue. In section IV, we will provide an even more efficient algorithm using only 1 set of tandem activated antennas.)

With the $|h_i|$ and $\theta_{i0}$, the channel phase difference between $h_i$ and $h_0$, for each antenna $i$ computed, the beamformer weights are set as $w_i = \frac{|h_i|}{\sqrt{\Sigma_{l=0}^{n-1}|h_l|^2}} e^{-\mathbf{j}\theta_{i0}}$ for $i > 0$ and $\mathbf{j} = \sqrt{(-1)}$, and $w_0 = \frac{|h_0|}{\sqrt{\Sigma_{l=0}^{n-1}|h_l|^2}}$.

### C. Optimality of RSSI based Beamforming

RSSI based beamforming is provably optimal (same performance as traditional beamforming). According to equation (1), the optimal beamforming setting should be $\mathbf{w} = \frac{1}{\sqrt{\Sigma_{l=0}^{n-1}|h_l|^2}}\mathbf{h}^*$. In our setting, the phase of $w_i$ is set to $-\theta_{i0}$, which is $-\arg(h_i) + \arg(h_0)$ rather than $-\arg(h_i)$; the phase of $w_0$ is set to 0 rather than $-\arg(h_0)$. This alternative setting only introduces a phase shift of $\arg(h_0)$ in $\mathbf{w}$, and thus will not affect the received signal power. Thus, this alternative setting also achieves the same optimal performance, as is proved in [2].

### D. Practical Advantages of RSSI based Beamforming

RSSI based beamforming is optimal, and also has several practical advantages: (i) full CSI approaches require specialized clients with the capability to measure the amplitude and phase of the received signals; RSSI based beamforming on the other hand can be implemented with off-the-shelf clients, (ii) oscillator related hardware synchronization impairments tend to corrupt the estimated CSI; RSSI based beamforming, on the other hand, solve the synchronization impairments by using the differential phase mechanism (using $-(\arg(h_i) - \arg(h_0))$ rather than $-\arg(h_i)$) [7], and (iii) full CSI state based beamforming approaches incur considerable overheads in the feedback sent to the transmitter; RSSI based beamforming on the other hand incurs trivial overheads as only RSSI values need to be communicated.

## III. Problem Motivation

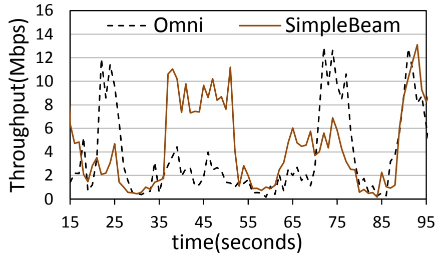
We use $SimpleBeam$ as the representative solution for RSSI based beamforming. Although $SimpleBeam$ is Optimal and has demonstratively better performance under static channel conditions[2], its performance under varying channel conditions (when $\mathbf{h}$ varies) has not been studied before.

We now present results from experiments that study the performance of $SimpleBeam$ when $\mathbf{h}$ varies. The experimental setting is described in Sec. VI-A. We consider mobile environments: (i) when the client orientation changes over time (Rotation), (ii) when there is human movement interfering with line of sight (LOS Blocking) periodically, and (iii) when the client is moving (Mobility). Under these environments, the variation of $\mathbf{h}$ becomes larger due to the large change in multipath fading conditions. Table 1(a) shows the degradation of throughput performance of $SimpleBeam$ in comparison to that in a static environment (when $\mathbf{h}$ is more static). Figure 1(b) shows that under certain conditions, $SimpleBeam$ can perform even worse than Omni. These results show that $SimpleBeam$ doesn't perform well when channel is time varying.

We contend that the underlying reason for the poor performance under channel varying conditions is the high time

| Rotation | | LOS blocking | | Mobility | |
|---|---|---|---|---|---|
| degree/s | % | Freq(1/s) | % | m/s | % |
| 180 | -64.84 | 1/10 | -42.07 | 0.5 | -37.28 |
| 360 | -88.24 | 1/5 | -71.44 | 1.5 | -58.49 |

(a) Degradation percentage




(b) Comparison with Omni under 1.5m/s mobility

Fig. 1. $SimpleBeam$ performance degradation in mobile environment

complexity of $SimpleBeam$. In order to gain optimal performance, $SimpleBeam$ needs to adjust the beamformer $\mathbf{w}$ according to the channel gains $\mathbf{h}$. When the rate of change of $\mathbf{h}$ becomes faster than the convergence time of $SimpleBeam$, the estimation of $\mathbf{h}$ will be erroneous, which causes significant performance degradation.

In Figure 2, we show an analysis of the phase fluctuation of $\mathbf{h}$ observed in two experimental setups. We see that the channel phase fluctuation can be as large as 100 degrees within a time span of tens of seconds with LOS blocking every 5 seconds, while it is only 30 degrees in a static environment. If the algorithm starts when $t = 50$ and finishes when $t = 70$, there will be a phase error $\theta_e = 100$ with LOS blocking every 5 seconds; the maximum $\theta_e$ is 30 degree in a static environment. Using equation (2) and assuming $P_i = P_j$, we can estimate the degradation percentage with phase error $\theta_e$ using equation $\frac{P_{\theta_e} - P_{opt}}{P_{opt}} = \frac{P_i+P_j+2\sqrt{P_iP_j}\cos\theta_e - P_i - P_j - 2\sqrt{P_iP_j}}{P_i+P_j+2\sqrt{P_iP_j}}$ $= \frac{1}{2}(\cos\theta_e - 1)$. Accordingly, a 30 degrees phase error will result in only $-6.69\%$ degradation, while a 100 degrees phase error will result in $-58.68\%$. Thus, the performance degradation will be significant if $\mathbf{h}$ differs a lot by the time the computation results are applied.

With $n$ antenna elements, $SimpleBeam$ relies on $n$ single antenna channel estimations, $n-1$ tandem antenna channel estimations, and an additional $4(n-1)$ tandem antenna estimations to solve the ambiguity due to the $\cos^{-1}$ function. Thus, the time complexity is $(6n-5)$ChannelEstTime, where ChannelEstTime is the time of channel estimation. In our experimental setup with 8 antenna elements, $SimpleBeam$ takes 22 to 27 seconds to perform its channel estimation (an implementation of the mechanism described in [2]). The large time complexity of $SimpleBeam$ thus is a drawback as channel gains $\mathbf{h}$ are likely to be different by the time algorithm finishes computing.

## IV. FastBeam Design

In this section, we present the key design elements of $FastBeam$, a set of four techniques that address the drawbacks discussed in Section III by reducing the time complexity. The first technique is optimality preserving, while the other three techniques are heuristic strategies toward reducing time complexity.

### A. *FewerPhases (optimality preserving)*

Based on mathematical analysis, we argue that the time complexity of $SimpleBeam$ could be reduced while still pre-

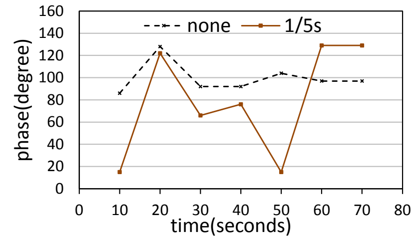



Fig. 2. Channel phase fluctuation with LOS blocking

serving its optimality. In $SimpleBeam$, 4 channel estimations are carried out for each channel phase difference $\theta_{ij}$ in order to solve the ambiguity due to the $\cos^{-1}$ function. This results in a time complexity of $4(n-1)$ChannelEstTime when there are $n$ antennas. Here we prove that the ambiguity can be resolved using only one estimation for each channel phase difference $\theta_{ij}$. We implement this concept in the algorithm $FewerPhases$.

*1) **Description**:* Instead of carrying out 4 channel estimations to solve the ambiguity, $FewerPhases$ uses only 1 tandem activated channel estimation to find $\theta_{ij}$ and sets $\mathbf{w}$ accordingly. Let $\rho_{ij}$ be the antenna phase difference of tandem activated antenna $i$ and $j$. Consider two cases:

Case 1 : If $\phi_{ij} \leq \frac{\pi}{2}$, set $\rho_{ij} = \phi_{ij}$.
Case 2 : If $\phi_{ij} > \frac{\pi}{2}$, set $\rho_{ij} = \pi - \phi_{ij}$.
If $P_{ij\_\rho_{ij}} \leq P_{ij}$, $\theta_{ij} = \phi_{ij}$; else, $\theta_{ij} = -\phi_{ij}$.

*2) **Proof of Optimality**:* According to [2], $\theta_{ij}$ could be $\phi_{ij}$, $-\phi_{ij}$, $\pi - \phi_{ij}$, or $-\pi + \phi_{ij}$. However, since it is possible to exactly know whether the $\cos(\theta_{ij})$ is positive or negative by calculating whether $(P_{ij} - P_i - P_j)$ is positive or negative, there are only two possibilities: $\theta_{ij} = \phi_{ij}$ or $\theta_{ij} = -\phi_{ij}$.

In Case 1, the received power will be as follows:

$$P_{ij\_\rho_{ij}} = P_i + P_j + 2\sqrt{P_iP_j}cos(\theta_{ij} + \phi_{ij}). \qquad (5)$$

i) If $\theta_{ij} = \phi_{ij}$, we will have $P_{ij\_\rho_{ij}} = P_i + P_j + 2\sqrt{P_iP_j}cos(\theta_{ij}+\theta_{ij})$, and since $\cos(2\theta) \leq \cos(\theta)$ when $|\theta| \leq \frac{\pi}{2}$, compared to equation 2, we have $P_{ij\_\rho_{ij}} \leq P_{ij}$.
ii) On the other hand, if $\theta_{ij} = -\phi_{ij}$, we will have $P_{ij\_\rho_{ij}} = P_i + P_j + 2\sqrt{P_iP_j}cos(\theta_{ij} - \theta_{ij})$, and since $\cos(0) \geq \cos(\theta)$, we have $P_{ij\_\rho_{ij}} \geq P_{ij}$.

In Case 2, following the similar deduction in Case 1, we have $P_{ij\_\rho_{ij}} \leq P_{ij}$ when $\theta_{ij} = \phi_{ij}$, and $P_{ij\_\rho_{ij}} \geq P_{ij}$ when $\theta_{ij} = -\phi_{ij}$.

*3) **Reduced Complexity**:* The time complexity of dealing with ambiguity in $FewerPhases$ is thus reduced to $(n-1)$ChannelEstTime. Thus, the total time complexity for $FastBeam$ is reduced to $(3n-2)$ChannelEstTime.

### B. *FewerAntennas (heuristic)*

When measuring $\mathbf{h}$, it can be observed a large diversity in the magnitude of channel gains across antennas. Since the time complexity of calculating the optimal beamformer weight of each channel is the same, it is reasonable to stop using the antenna corresponding to the channel with poor magnitude and hence save on the time complexity. $FewerAntennas$ adaptively reduces the number of antennas used, and hence reduces the time complexity.

*1) **Description**:* $FewerAntennas$ first performs single channel estimation and gets the $P_i$ for each antenna, and sorts them according to their power. Without loss of generality, we assume that $P_0 \geq P_1 \geq P_2 \cdots \geq P_n$. Then $FewerAntennas$

determines the number of antennas to use, $n_{ant}$, using the following equation:

$$n_{ant} = \arg_k \min(\Sigma_{i=0}^{i=k} \sqrt{P_i} \geq \alpha \Sigma_{i=0}^{i=n} \sqrt{P_i}), \qquad (6)$$

where $\arg_k \min(Statement)$ returns the smallest $k$ that satisfy the *Statement*, and $\alpha < 1$ is a threshold which guarantees that the received power is at least a fraction $\alpha$ of the total power.

After determining the number of in-use antennas $n_{ant}$, the rest of the $FastBeam$ algorithm is same as that of $FewerPhases$, but with a reduced set of antennas.

*2) **Justification**:* Although using fewer antennas will decrease the maximum possible received power, the reduction in time complexity can reduce the estimation error for $\mathbf{h}$, which results in better performance. As mentioned in section III, $\theta_e = 30$ will result in only $-6.69\%$ degradation from optimal received power, while $\theta_e = 100$ will result in $-58.68\%$. This leads to a large potential to increase throughput by sacrificing the maximum possible received power and decreasing the computational time complexity for more accurate estimation of $\mathbf{h}$. If $\alpha = 0.8$, and due to the decrease in time complexity we could have $\theta_e \leq 30$, the degradation of $FewerPhases$ is at most $-24.80\%$; compared to the possible $-58.68\%$ degradation in $SimpleBeam$, the received power of $FewerAntennas$ is $1.8\times$ better.

*3) **Reduced Complexity**:* The time complexity of tandem activation and ambiguity resolution are both reduced to $(n_{ant} - 1)$ChannelEstTime. Thus, the total time complexity is reduced to $(n + 2n_{ant} - 2)$ChannelEstTime.

### C. ZeroPhase (heuristic)

Although $FewerAntenna$ decreases the time complexity by a certain amount, in order to deal with environments having high variance of channel conditions, it is desirable to have an algorithm that has even less time complexity, at the cost of accuracy. $ZeroPhase$ is a heuristic algorithm that attempts to construct the beam pattern without even computing the exact channel phase $\theta_{ij}$. It simply controls the signal on each antennas to be constructive rather than destructive to the reference antenna (i.e. antenna 0).

*1) **Description**:* $ZeroPhase$ first carries out the single and tandem activated channel estimations to get $P_i$ and $P_{i0}$ for each antenna. Then, for each antenna $i$, $i \neq 0$, if $P_i + P_0 > P_{i0}$, the phase of $w_i$ is set as $\pi$; else, the phase of $w_i$ is set as 0.

*2) **Justification**:* Since $P_{i0} = P_i + P_0 + 2\sqrt{P_i P_0} cos(\theta_{i0})$, if $|\theta_{i0}| > \frac{\pi}{2}$, we will have $P_i + P_0 > P_{i0}$. By setting the phase of $w_i$ as $\pi$, the received power of the two antenna will become $P_{i0} = P_i + P_0 + 2\sqrt{P_i P_0} cos(\theta_{i0} + \pi)$, and since $\cos(\theta_{i0} + \pi) > 0$, which will make $P_i + P_0 < P_{i0}$, the destructive effect of the two signals becomes constructive. On the other hand, if we already have $P_i + P_0 < P_{i0}$, we will simply set the phase of $w_i$ as 0 to keep it constructive.

Further, we use the equation: $\frac{1}{2}(\cos\theta - 1)$ from section III to estimate the degradation of $ZeroPhase$. Since $ZeroPhase$ always control the signal to be constructive, the $\theta_e$ is at most 90 degree: the degradation is at most $\frac{1}{2}(\cos\frac{\pi}{2} - 1) = -50.00\%$; compared to the possible $-58.68\%$ degradation in $SimpleBeam$, the received power of $ZeroPhase$ is $1.2\times$ better in the worst case.

*3) **Reduced Complexity**:* Since $ZeroPhase$ only needs to carry out the single and tandem activated channel estimations, the total time complexity is reduced to $(2n - 1)$ChannelEstTime.

### D. FingerPrint (heuristic)

Beyond the strategies discussed thus far, with historical information about the environment, the time complexity can be reduced even further. $FingerPrint$ is a heuristic algorithm that uses the pre-measured, recorded (fingerprint $\mathbf{f_k}$, beamformer $\mathbf{w_k}$) pairs to set up the beamformer $\mathbf{w}$. The fingerprint $\mathbf{f}$ is a sequence of received power values that reflect the characteristics of the channel gain $\mathbf{h}$. The algorithm measures the fingerprint $\mathbf{f}$ of the current channel gain $\mathbf{h}$, compares it with previously recorded fingerprints $\mathbf{f_k}$, and then uses the corresponding recorded beamformer $\mathbf{w_k}$ if there is a match in the fingerprint.

*1) **Description**:* $FingerPrint$ will first measure the fingerprint $\mathbf{f} = (P_0, P_{01}, P_{02}, \cdots, P_{0n})$ of the current channel gain $\mathbf{h}$, and search among its data base of fingerprints ($\mathbf{f_k}$, $\mathbf{w_k}$). If there is a match, i.e. there is a $\mathbf{f_k}$ having similar characteristic as $\mathbf{f}$, $FingerPrint$ sets the beamformer as $\mathbf{w_k}$; otherwise the algorithm will run $FewerPhase$ and record the resulting fingerprint and beamformer pattern ($\mathbf{f}$, $\mathbf{w}$). The matching function is: $|\mathbf{f} - \mathbf{f_k}|$. If the value of matching function is smaller than a threshold $\beta$, there is a match.

*2) **Justification**:* Each channel gain $\mathbf{h}$ will have its own fingerprint $\mathbf{f}$. If $\mathbf{h} \neq \mathbf{h}'$, the probability that $\mathbf{f} = \mathbf{f}'$ is very small. Since the fingerprint $\mathbf{f}$ contains the power level of reference antenna $P_0 = |h_0|^2$ and the $P_{0i} = |h_0 + h_i|^2$, $\mathbf{f}$ captures the channel phase difference $\theta_{0i}$ between $h_0$ and $h_i$. Assuming there are $k_p$ possible level of channel power gain $|h_i|$ and 360 possible degree of channel phase $\arg(h_i)$. That is, there are $k_p \times 360$ possible value of $h_i$. According to equation (3), given values of $P'_{0i}$, $P_0 = P'_0$, and $P_i = |h_i|^2$, only two value of $\theta_{0i}$ will make $P_{0i} = P'_{0i}$. Assuming uniform distribution, $Pr(P_0 = P'_0|P'_0) = \frac{360}{k_p \times 360} = \frac{1}{k_p}$, $Pr(P_{01} = P'_{01}|P'_{01}, P_0 = P'_0, h_1 \neq h'_1) = \frac{2 \times k_p - 1}{360 \times k_p} \cong \frac{1}{180}$. Thus, $Pr((P_0, P_{01}) = (P'_0, P'_{01})|P'_{01}, P'_0, (h_0, h_1) \neq (h'_0, h'_1)) \cong \frac{1}{180 \times k_p}$, and $Pr(\mathbf{f} = \mathbf{f}'|\mathbf{f}', \mathbf{h} \neq \mathbf{h}') \cong \frac{1}{180^{(n-1)} \times k_p}$. If $n = 8$, $k_p = 10$, $Pr(\mathbf{f} = \mathbf{f}'|\mathbf{f}', \mathbf{h} \neq \mathbf{h}') < 10^{-15}$.

Therefore, if we measured the current fingerprint $\mathbf{f}$ and infer that $\mathbf{f} = \mathbf{f}'$, there is a high probability that $\mathbf{h} = \mathbf{h}'$. Thus, if we have the recorded pair ($\mathbf{f}'$, $\mathbf{w}'$), the recorded beamformer $\mathbf{w}'$ can be applied.

Since there always exists a small variance in the measurement values of power, $FingerPrint$ does not strictly require that $\mathbf{f} = \mathbf{f_k}$. Thus, as long as the difference $|\mathbf{f} - \mathbf{f_k}|$ is smaller than a preset threshold, $FingerPrint$ infers them to be equal.

*3) **Reduced Complexity**:* Since $FingerPrint$ only needs to carry out one single activation and the tandem activated channel estimation, the total time complexity is reduced to $n$ChannelEstTime if there is a match. If there is no match, the time complexity will be the same as $FewerPhase$.

**Algorithm 1** FastBeam algorithm

INPUT:
$V$ = variable indicating the level of RSSI variance at client. The value of $V$ could be "Low", "Medium", or "High".
ALGORITHM:

```
1: for each update cycle or request for update from client do
2:     f = Fingerprint_Measurement();
3:     w = Recorded_Pattern.match(f);
4:     if w != NULL then
5:         jump to line 14;
6:     else if V == "Low" then
7:         w = FewerPhase(f);
8:         Recorded_Pattern.record(f,w);
9:     else if V == "Medium" then
10:        w = FewerAntennas(f);
11:    else if V == "High" then
12:        w = ZeroPhase(f);
13:    end if
14:    set up w;
15: end for
```

## V. FastBeam Solution Details

We now present the details of $FastBeam$ solution composed of the 4 algorithms described in Section IV. We consider one access point(AP) equipped with multiple-antennas and one off-the-shelf client with omni antenna. The pseudo code for $FastBeam$ is presented in Algorithm 1.

$FastBeam$ is a two-ended algorithm that requires client participation. However, the client participation is minimal and is purely in software. Since it is necessary to carry out smaller time complexity algorithms when channels change rapidly, $FastBeam$ selects among the 4 algorithms based on the variance of RSSI measured by the client. Based on the RSSI records over time, the client reports the rate of change of its RSSI values using a three-level indicator: "High", "Medium", and "Low". The level of variance $V$ is sent to the AP periodically or when the client wants to trigger an update of the beamformer $\mathbf{w}$ if the RSSI is smaller than a threshold.

The AP performs the $FastBeam$ algorithm periodically or when it receives an explicit request from the client. Since we always have very good performance when there is a match in $FingerPrint$, here we design and implement $FastBeam$ to use the recorded beamformer $\mathbf{w}$ whenever there is a match. A more careful design could carry out a RSSI measurement for the matched beamformer $\mathbf{w}$ to see if the performance is satisfiable, and decide not to use the matched beamformer $\mathbf{w}$ if the measured RSSI is low.

Algorithm 1 combines the four algorithms in a way that optimally utilizes the common channel estimation values among the four algorithms (lines 7, 10, and 12 utilize the channel estimation values in $\mathbf{f} = (P_0, P_{01}, P_{02}, \cdots, P_{0n})$). $FastBeam$ first measures the fingerprint $\mathbf{f}$ of the current channel, and sees if there is a match (lines 1-3). If there is a match, it uses the corresponding beamformer $\mathbf{w}$ (lines 4 to 5). If there is no match in $FingerPrint$ (or the matched beamformer $\mathbf{w}$ is not satisfiable),an algorithm among the rest three will be selected according to the level of the variance $V$ reported by the client. If $V$ is "High", the base station performs the $ZeroPhases$ algorithm, which has the lowest time complexity (line 12); if $V$ is "Medium", the base station performs the $FewerAntennas$ algorithm (line 10); and if $var$ is "Low", the base station performs the $FewerPhases$ algorithm. When $FewerPhases$ is carried out, it records the resulting beamformer $\mathbf{w}$ with the measured fingerprint $\mathbf{f}$ (lines 7 and 8). The database of $FingerPrint$ can be set up previously by carrying out dedicated training or purely learned during the operation. The timely learning of the fingerprint database make $FastBeam$ adaptive to the environment.

## VI. Performance Evaluation

In this section, we use implementations of the $FastBeam$ and $SimpleBeam$ algorithms on an experimental multi-antenna AP platform called Phocus Array([3]) for performance comparison.

### A. Experimental Setting

We implement the $FastBeam$ and $SimpleBeam$ algorithms on the Phocus Array platform that has 8 programmable antennas. The Phocus Array platform acts as an AP, and the server(controller) is a PC that is directly connected to the AP. An off-the-shelf client wirelessly connects to the AP. We compare the throughput performance using iperf sessions between the client and the server under different situations.

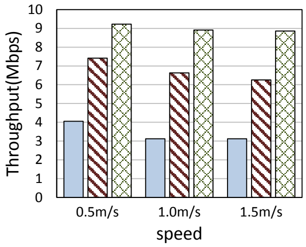
The CPU of the AP is XScale-Ixp42x with BogoMIPS equal to 532.48, and the CPU of the server is Intel Pentium 4 2.80GHz; the memory of the AP and the server are 128MB and 1GB, respectively.

The experiment is carried out in a typical office environment, where there are walls and furniture. Since the possible movement of a client is composed of straight movement and rotation, we set up experiments when the client is moving in a straight line and when it is rotating. Also, since the change in the environment is typically caused by people, we set up experiments when there are people crossing the LOS between the client and the AP periodically. Most of the indoor environments are combinations of these situations.

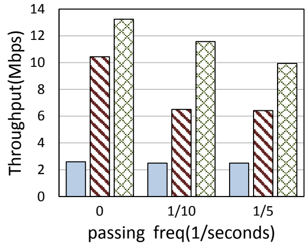
### B. Macro Performance

Fig. 3(a) shows the throughput of the Omni, $SimpleBeam$, and $FastBeam$ algorithms under different rates of client mobility. In this scenario, the client moves at a certain speed along a line trajectory back and forth. $FastBeam$ outperforms $SimpleBeam$ due to the decrease in time complexity. The improvement increases with the mobility speed (from $1.24\times$ to $1.42\times$).

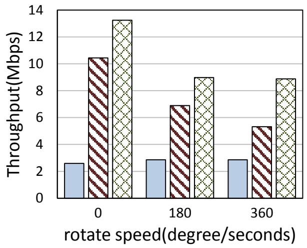
Fig. 3(b) shows the throughput of the 3 algorithms under different frequencies of LOS blocking by an interfering object. In this scenario, the client is static, and there is a human passing through the LOS between the client and the AP every few seconds. $FastBeam$ again outperforms $SimpleBeam$, and the improvements are from $1.27\times$ to $1.78\times$. Note that $FastBeam$ performs better even when the passing frequency



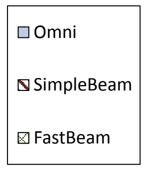

(a) Client mobility




(b) LOS blocking

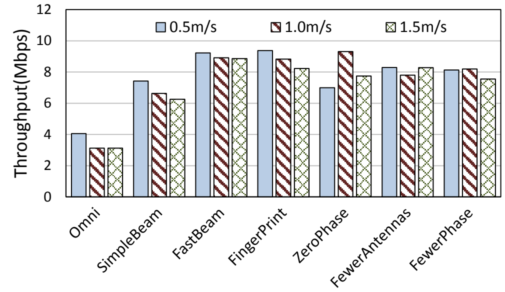



(c) Client rotation

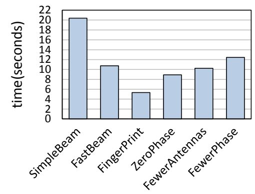



(d) Figure label

Fig. 3. $FastBeam$ performance




(a) Throughput performance




(b) Convergence time

| Algorithm | Memory(%) | | CPU(%) | |
|---|---|---|---|---|
| | AP | Srv | AP | Srv |
| SimpleBeam | 0.53 | 0.66 | 1.52 | 0.09 |
| FastBeam | 0.53 | 0.64 | 0.78 | 0.05 |
| FingerPrint | 0.53 | 0.61 | 0.49 | 0.03 |
| ZeroPhase | 0.53 | 0.61 | 0.26 | 0.02 |
| FewerAntennas | 0.53 | 0.63 | 0.48 | 0.03 |
| FewerPhase | 0.53 | 0.64 | 0.59 | 0.03 |

(c) Memory and CPU usage

Fig. 4. Performance of each algorithm

is 0. This is because $\mathbf{h}$ has small changes even under static environment, which could be observed in fig. 2. Thus, $FastBeam$ could get benefit from its smaller time complexity even when the environment is static.

Fig. 3(c) shows the throughput of the 3 algorithms under different rates of client rotation. In this scenario, the client is put on a platform, and it rotates at a fixed speed. $FastBeam$ outperforms $SimpleBeam$, and the improvement increases with the rotation speed (from $1.27\times$ to $1.67\times$).

As shown in Fig. 3, $FastBeam$ outperforms $SimpleBeam$ under each situation due to the adaptive selection among the 4 algorithms that could lead to a good trade-off between time-complexity and accuracy.

### *C. Micro Performance*

We now present the throughput performance for the four individual algorithms, which together compose the $FastBeam$ algorithm. As can be seen in Fig. 4(a), $FewerPhases$ performs better than $SimpleBeam$ in every environment. $FingerPrint$ performs very well when there is a match, and the performance reduces to the same level as $FewerPhases$ when there is no match. Thus, overall $FingerPrint$ performs better than $FewerPhases$, and we can see that the fingerprint matching strategy does work.

The performance of $FewerAntennas$ is rather dynamic because it depends more on the characteristics of the amplitude of the channel gain $\mathbf{h}$ than the rate of client mobility: If the amplitude of the channel gain $\mathbf{h}$ is much larger on certain channels, $FewerAntennas$ can achieve close to the optimal received power with much less time complexity, which leads in better improvement; on the other hand, if the amplitude of the channel gain $\mathbf{h}$ is almost equal on each channels, $FewerAntennas$ will have less improvement. The $ZeroPhase$ performance is even more dynamic, since its performance heavily depends on the phase difference $\theta_{ij}$ of the channel gain $\mathbf{h}$. If $\theta_{ij}$ is closer to 0 or $\pi$, $ZeroPhase$ will perform much better; on the other hand, if $\theta_{ij}$ is close to $\frac{\pi}{2}$, $ZeroPhase$ will perform sub-optimal. However, because of the reduction in time complexity, $ZeroPhase$ has better performance than $SimpleBeam$ when the moving speed becomes larger than 1.0m/s.

Figure 4(b) shows the time required to perform each algorithm. As can be seen, combining the different times of the four algorithms, $FastBeam$ is about 2x faster than $SimpleBeam$ on average and 4X when there is a match.

### *D. Overheads/Complexities*

Table 4(c) shows the CPU and memory usage of each algorithm on AP and the server (Srv). As can be seen, the CPU and memory usage of all algorithms are both very low. The CPU usage is below 2%, and the memory usage is below 1% for the platforms used in the experiments.

## VII. Conclusions

In this work, we explore algorithms to perform *fast beamforming*. While existing techniques for RSSI based beamforming severely under-perform when the channel changes rapidly, $FastBeam$ could dynamically adapt with the variation of the channel and still retain the advantages of RSSI based beamforming. It only requires software modification on the client, thus is easy to deploy. Experimental evaluation shows that the throughput performance of $FastBeam$ is $1.4\times$ in average and up to $1.8\times$ in the best case, compared to the existing algorithm.

## References

[1] A. Paulraj, R. Nadar, and D. Gore, "Introduction to space-time wireless communications," *Cambridge University Press*, May 2003.

[2] S. Lakshmanan, K. Sundaresan, S. Rangarajan, and R. Sivakumar, "Practical beamforming based on rssi measurements using off-the-shelf wireless clients," *ACM IMC*, 2009.

[3] "Phocus array: Fidelity-comtech Inc." [Online]. Available: http://www.fidelity-comtech.com.

[4] S. Lakshmanan, K. Sundaresan, S. Rangarajan, and R. Sivakumar, "The myth of spatial reuse with directional antennas in indoor wireless networks," *PAM*, 2010.

[5] X. Liu, S. Seshan, and P. Steenkiste, "When are directional antennas useful in indoor environments?" *WiNECH*, 2011.

[6] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "Dirc: Increasing indoor wireless capacity using directional antennas," *ACM SIGCOMM*, 2009.

[7] S. Lakshmanan, K. Sundaresan, R. Kokku, A. Khojestepour, and S. Rangarajan, "Towards adaptive beamforming in indoor wireless networks: An experimental approach," *IEEE INFOCOM (Miniconference)*, 2009.