Sensor-assisted Codebook-based Beamforming for Mobility Management in 60 GHz WLANs

Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Prasant Mohapatra
Department of Computer Science
University of California, Davis, CA, USA.
Email: {zcyang, phpathak, zeng, pmohapatra}@ucdavis.edu

***Abstract*—The potential to provide multi-gbps throughput has made 60 GHz communication an attractive choice for next-generation WLANs. Due to highly directional nature of the communication, a 60 GHz link faces frequent outages in the presence of mobility. In this work, we present a sensor-assisted multi-level codebook-based beamwidth adaptation and beamswitching to address the mobility challenges in 60 GHz WLANs. First, we show that by combining antenna element selection with codebook design, it is possible to generate a multi-level codebook that can cover different beamforming directions with many possible beamwidths and directive gain. Second, we propose that accelerometer and magnetometer sensors which are commonly available on mobile devices can be used to better account for mobility, and perform near-real time beamwidth adaptation and beamswitching. We evaluate the sensor-assisted multi-level codebook-based beamforming with trace-driven simulations using real mobility traces. Numeric evaluation shows that such beamforming can maintain the connectivity over 84% of the time even in presence of high device mobility.**

I. Introduction

Recently, communication in 60 GHz unlicensed spectrum (57-66 GHz) has recently gained attraction due to its capability of providing multi-gbps throughput in WLANs. Various wireless standards such as IEEE 802.11ad [1] and its predecessor WiGig [2] have shown that 60 GHz links can provide nearly 7 Gbps of throughput for distances lesser than 10 meters. Such high throughput can enable numerous applications like wireless high-definition video streaming between devices (e.g. from smartphone to television) and wireless USB, HDMI and PCI-e interconnections.

However, the high throughput comes at a cost that 60 GHz communications are required to be highly directional (sector widths as low as 10 degrees). This is because signal propagation characteristics at 60 GHz frequency are noticeably different from 2.4/5 GHz. The path loss is nearly 22 dB higher at 60 GHz compared to 2.4/5 GHz (derived using Friis Free Space model [3] and observed in measurements [4]). To compensate for such high loss, an array of antenna elements is used to generate a high-gain directional beam towards the intended receiver. An antenna array at 60 GHz can have a large number of elements even with a small overall antenna size. This is because the distance between the elements is in millimeters (half the wavelength), allowing the antenna to fit in mobile devices such as smartphones. Antenna array allows digital beamforming that can adaptively switch beam without requiring any mechanical steering of antenna components.

Different beam patterns generated using an antenna array depend on the number of antenna elements, their arrangement, and phase shift weights applied to the elements. Due to a large number of antenna elements (typically required to achieve a reasonable gain), optimal beamforming towards the receiver is very expensive. The estimation overhead and complexity of the complete channel state information (CSI) matrix become impractical as the number of antenna elements increases. Recently, codebook-based beamforming has been proposed to eliminate such overhead. In a codebook, a set of beamforming weights (beam patterns) is pre-calculated, and a specific weight vector is applied depending on the desired beamforming direction and gain. Although not optimal, codebook-based beamforming is shown to achieve a fair balance between complexity and performance, and is also adapted in recent 60 GHz standards [1], [5].

In order to select a specific beam pattern from codebook, both endpoints perform an iterative search procedure referred as Sector Level Sweep (SLS). In this procedure, one endpoint (called beamformer) sends out one or more probe messages to the intended receiver (called beamformee) in all possible sectors. The beamformee measures the received signal strength for probe received in all different sectors. It then replies to the beamformer on the sector for which the signal strength of the received probe was the highest. Although this procedure is time consuming, it is required to be performed only once when the endpoints are stationary.

The problem is that when one or both endpoints are mobile, such beamforming is extremely challenging. In the face of mobility, the nodes have to constantly perform beam searching, requiring (possibly limited) SLS to re-establish the link. This kind of frequent interruptions and disconnections are not desirable especially in 60 GHz networks where most of the applications are likely to be multimedia (e.g. live video presentations from smartphone to projector). Fig. 1 shows an example scenario where one endpoint is stationary (e.g. TV, projector) and the other endpoint (e.g smartphone, tablet) moves around in close vicinity. A common application in such scenario is that a mobile device is streaming its screen to a larger display (the stationary node). In such case, both endpoints require constant beam searching towards each other in order to maintain a link. The time consumed by constant beam searching can reduce the link QoS significantly.

In this paper, we propose SAMBA (Sensor-Assisted Multi-level Codebook-based Beam Adaptation) to address the mobility challenge in 60 GHz beamforming. SAMBA is built

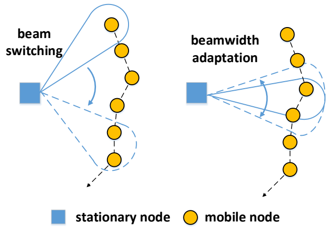


Fig. 1: Beam switching and beamwidth adaptation

on the idea that if the mobile device's movement can be predicted, efficient beamwidth adaptation and beam switching (combinedly referred as "beam adaptation") can be achieved even using codebook. For such efficient beam adaptation, SAMBA relies on two components - (i) a novel multi-level codebook that provides diverse choices of beamwidths in different beamforming angles and (ii) an anticipatory movement prediction algorithm that utilizes mobile device's sensors (accelerometer and magnetometer) to accurately forecast its next location. Combining the sensor-based mobility prediction with multi-level codebook, SAMBA can *proactively* adapt the beam to provide uninterrupted connections even in the presence of high mobility.

The contributions of this paper are as follows:

(1) We design a multi-level codebook that enables a wide variety of beamwidths and directive gain (also called directivity) for different beamforming directions. The multi-level codebook uses intelligent antenna selection to activate and deactivate certain subset of antenna elements depending on the desired characteristics of beam. We provide a generalized algorithm that to design a multi-level codebook for a given antenna array.

(2) We propose SAMBA which performs beam switching and beamwidth adaptation depending on client's mobility. SAMBA uses the information provided by mobile device's sensors (accelerometer and magnetometer) to estimate its mobility characteristics such as heading direction and speed. Based on the anticipated mobility, SAMBA can proactively adapt the beam to provide a robust link towards the mobile device. We evaluate SAMBA using real indoor mobility traces and trace-driven simulations. We observe that SAMBA can successfully provide coverage to the mobile client over 84% of the time even in the presence of high-speed and random mobility.

The rest of the paper is organized as follows. We first provide phased antenna array and codebook related background in Section II. The system model and problem description are provided in Section III. The design of multi-level codebook is discussed in Section IV and sensor-assisted beam adaptation algorithm is provided in Section V. We evaluate SAMBA in Section VI and conclude in Section VII.

II. Background and Related Work

In this section, we first provide some background on phased antenna array and codebook-based beamforming procedure.

A. Phased Antenna Array

Since the carrier wavelength is as small as a few millimeters for 60 GHz, it is possible to integrate multiple antenna elements in an array. Such an array is much smaller in size for 60 GHz as compared to an antenna array for 2.4/5 GHz which makes it feasible to use it in mobile devices (e.g. smartphones). The antenna array can be used to perform beamforming to enable directional communication with high directive gain. Different values of phase shifts are applied to each antenna element in order to establish a beam in a given direction. The direction of the beam and the beamwidth also depend on the arrangement of the antenna elements. As an example, Fig. 2a shows three different antenna element arrangements for a total of 16 elements - linear 1×16, rectangular 2×8, and square 4×4. Each pair of adjacent elements is spaced at a distance of half the wavelength. The antenna's directive gain in azimuth plane is shown in Fig. 2b and Fig. 2c shows the 3-D directive gain. As we can observe, the linear array (1×16) provides narrower beamwidths compared to the rectangular and the square arrays. Such narrower beamwidths introduce higher overhead to accurately point the beam towards the receiver and have less tolerance for receiver mobility. For these reasons, many current 60 GHz manufacturers (e.g. Wilocity chipsets [6]) use rectangular antenna array (such as 2×8) to achieve a balance between gain and tolerance for receiver mobility.

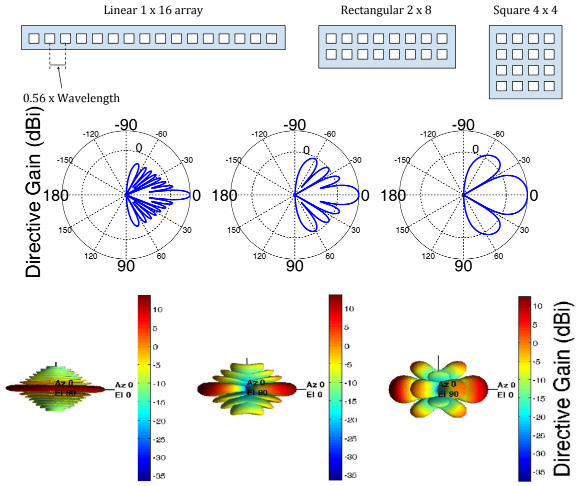


Fig. 2: (a) Linear, rectangular and square arrangement of 16 antenna elements (b) Directive gain in azimuth cut for the three arrangements and (c) Their 3D directive gain

B. Adaptive Beamforming vs. Codebook Design

There are two beamforming techniques that are actively studied for 60 GHz communication - adaptive beamforming and codebook-based switched beamforming.

1) Adaptive Beamforming: In adaptive beamforming, the transmitter utilizes the complete CSI and AoA (Angle Of Arrival) to calculate the beam direction and maximize the beamforming gain. The beamforming weights for antenna elements are calculated by the pseudo-inverse of the CSI matrix. Although such beamforming is shown to maximize the directive gain [7], [8], the computational complexity and overhead of CSI matrix makes it almost infeasible to use in practice. This is because the CSI overhead increases significantly with a large number of antenna elements as shown in [9]. This not only increases the beamforming setup time but even a minor change in wireless channel requires much more computation which does not scale in delay sensitive applications. This way, although adaptive beamforming is useful in 802.11n/ac like systems with fewer antennas, it becomes impractical to be used for large antenna arrays used in 60 GHz.

2) Codebook-based Beamforming: To address the problems of adaptive beamforming, a switched beamforming based on structured codebook was proposed in [10]. Here, different beamforming weights are precalculated in the form of a codebook, and different beam patterns are realized by switching/choosing the appropriate weight vector from the codebook. Because of its simplicity, less overhead and scalability, codebook-based beamforming has been widely adapted in standards [1], [5] as well. When utilizing codebook, the 60 GHz endpoints use SLS in order to iteratively *find* each other. In the SLS, first one endpoint (beamformer) sends training frames in all possible sectors (using different weights of codebook) while the other endpoint (beamformee) listens by configuring its antenna in quasi-omni pattern. Based on the feedback from the beamformee, the beamformer chooses the best beam/sector to communicate to the beamformee. This procedure is typically followed by *Beam Refinement Protocol* [1] which further improves the beamforming accuracy.

III. System Model and Problem Description

In this section, we describe the system model and the problem, and then outline our proposed solution.

A. System Model

We consider a system where both of the transmitter and the receiver are equipped with a 60 GHz radio and an antenna array. We assume the scenario where one endpoint is stationary (referred as AP) and the other endpoint is mobile (referred as client). This is a common scenario proposed in example 60 GHz use-cases where a smartphone streams its screen to a television set or a video projector. With some added intelligence, our beamforming solution can be extended to the case where both endpoints are mobile nodes. In our system, both endpoints form a beam towards each other to establish a bidirectional link. Since SAMBA uses sensors to aid the beamforming process during mobility, we assume that the mobile node is equipped with accelerometer and magnetometer sensors. Let us assume that the transmitter and the receiver are equipped with N_t and N_r antenna elements respectively. Like most 60 GHz radio designs [7], we assume that the beamforming happens in the RF domain using a single RF chain. This only requires one ADC/DAC eliminating the energy and computational overhead of using multiple ADC/DACs. As shown in Fig. 3, transmit beamforming is achieved via applying the transmit weight vector $\{w_i\}_{i=1}^{N_t}$ to the RF signal, while at the receiver, the receive weight vector is applied on the received RF signal and then the signal is converted to the baseband.

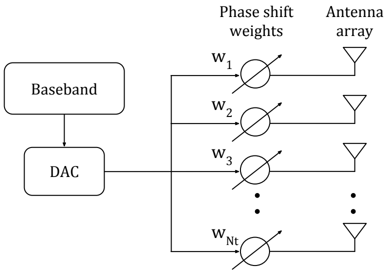


Fig. 3: Transmit beamforming using phase shift weights and antenna array elements

$$\begin{array}{c} \\ idx1 \\ idx2 \\ idx3 \\ idx4 \end{array}\begin{array}{c} \begin{array}{cccc} ptn1 & ptn2 & ptn3 & ptn4 \end{array} \\ \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2}i & -\frac{1}{2} & \frac{1}{2}i \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2}i & -\frac{1}{2} & -\frac{1}{2}i \end{bmatrix} \end{array}$$

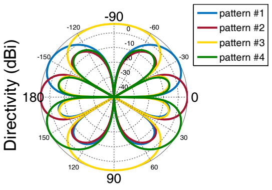


Fig. 4: Codebook weight matrix and directive gain of resultant patterns for a 1 × 4 antenna array. The array element indexes are left to the codebook and the pattern indexes are above it

B. Codebook Design

Our beamforming technique SAMBA assumes codebook-based beamforming on both endpoints. In simpler terms, a codebook [7], [10] is a collection of beams where each beam covers a specific direction in space and all beams collectively exhaust the entire space. Formally, a codebook is a $P \times Q$ matrix where P is the number of antenna elements and Q is the number of weight vectors each of which generates a specific beam pattern. In this work, we use the DFT-based codebook design as proposed in [11] as it is proven to achieve a uniform gain in all directions. The DFT-codebook matrix can be calculated using -

$$w(p, q) = \frac{1}{\sqrt{P}} e^{-j2\pi(p-1)(q-1)/Q} \tag{1}$$

where $p \in \{1, ..., P\}$, $q \in \{1, ..., Q\}$ and $j = \sqrt{-1}$.

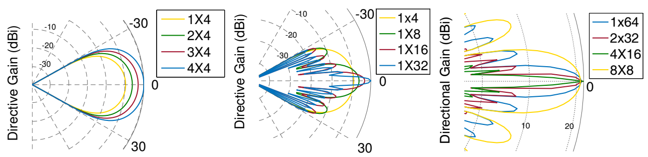
Fig. 4 shows an example of a codebook weight matrix for a 1 × 4 linear antenna array. It also shows the directive gain of the four patterns in the azimuth plane.

C. Problem Description

Although the codebook-based beamforming works efficiently in the beam setup process (using SLS), it assumes that the endpoints remain stationary after the beamforming process is complete. This is rarely the case in real-world applications. In a typical use-case, the client will move around the AP requiring frequent changes in the beam. The limitation of codebook-based beamforming is however that if there is a significant change in the client's location, the current beam pattern is rendered useless and a new beam has to be established towards the new location of the client. The beam searching procedure typically requires a complete or partial SLS which essentially interrupts the ongoing communication. The delay of beam searching is reported to be as high as 5-6 seconds in current 60 GHz hardware [12]. Our objective is to design a beam adaptation technique that can retain the efficiency and scalability characteristics of codebook while keeping a robust link even in the presence of client mobility.

D. Approach

To address the mobility challenges of codebook-based beamforming, we investigate the answer to the following



(a) (b) (c)

Fig. 5: Impact of activating different number of rows, columns, and different arrangements of elements in an antenna array

question: Today's mobile devices like smartphone have accelerometer and magnetometer sensors. Can the AP use the client's sensor observations to predict its mobility and use this prediction to proactively adapt the beam? As we show, the answer is affirmative. In order for the AP to proactively adapt the beam, there are two requirements -

(1) An enhanced codebook that can provide a variety of beamwidth choices. Here, if the client mobility is high, larger beamwidth can be used to provide a robust coverage. On the other hand, if the client mobility is low, narrow beamwidth with higher gain can be used. We propose a multi-level codebook in Section IV which utilizes the antenna selection to generate different beams using the same antenna array.

(2) A method that can predict the mobile device's next location given its sensor observations. We show in Section V that accelerometer and magnetometer observations can be combined to find smartphone's heading direction and traveled distance or speed. Using this information, SAMBA can predict smartphone's movement and proactively switch the beam across the multi-level codebook.

We primarily focus on AP to client beamforming as it is significantly more challenging. In Section V, we describe how SAMBA can be used for client to AP beamforming.

IV. Multi-Level Codebook Design

The conventional codebook design generates beams with identical amplitude, and assumes that all antenna elements of the array are active when generating different beam patterns. This means that it is not possible to cover any beamforming angle with more than one beamwidth and amplitude. As we discussed before the usage of conventional codebook in the presence of receiver mobility results in frequent outages. While both 802.15.3c and 802.11ad theoretically support the multi-level codebook, few beam choices are available in terms of beam beamwidth and amplitude [7]. In this section, we propose that when the *antenna elements selection* is combined with the codebook design, it is possible to generate different beam patterns that can cover many different beamforming directions with multiple beamwidths and amplitudes. In theory, given an antenna array with M rows and N columns, a separate codebook can be generated for different subsets of $m \times n$ where $m \subset M$ and $n \subset N$. Since there can be a large number of such possible subsets, first it is necessary to understand which subsets are actually useful in generating different beamwidths and patterns. A separate codebook is then generated for each of such subset and the joint codebook of all subsets is referred as the multi-level codebook.

Algorithm 1 Multi-level Codebook Generation

```
Input:
    An antenna array $\mathbb{A}$ of size $M \times N$ where $M$ is the number of rows and $N$ is the number of columns, $M = 2^r$, $N = 2^c$, $r \geq 0$, $c \geq 1$
Output:
    A multi-level codebook matrix $\mathcal{C}$ of size $P \times Q$ where $P = M \cdot N$ and $Q$ is the number of patterns
Procedure:
    Build vectors $V_R$ and $V_C$ such that $V_R = [1, 2, 4, ..., 2^r]$, and $V_C = [1, 2, 4, ..., 2^c]$
    for each unique combination of $v_r \in V_R$ and $v_c \in V_C$, create an antenna array subset $\mathbb{S}^{v_r, v_c}$ do
        # Generate the DFT-based codebook $\mathbf{X}^{v_r, v_c}$ of
        # subset $\mathbb{S}^{v_r, v_c}$ where the number of patterns = $v_c$
        # and the number of elements = $v_r \cdot v_c$
        for $q$ = 1: $v_c$ do
            for $p$ = 0: $(v_r \cdot v_c) - 1$ do
                $\mathbf{X}^{v_r, v_c}(p+1, q) = \frac{1}{\sqrt{v_c}} e^{-j 2\pi (p \bmod v_c)(q-1)/v_c}$
            end for
        end for
    end for
    # Append the subset codebook to the multi-level
    # codebook while preserving the weights and
    # arrangement of the subset
    Map the subset $\mathbb{S}^{v_r, v_c}$ to $\mathbb{A}$ while preserving the arrangement of $v_r \times v_c$
    Append all the patterns (columns) of $\mathbf{X}^{v_r, v_c}$ to $\mathcal{C}$, set the weights to be 0 for all the elements not in $v_r \times v_c$
```

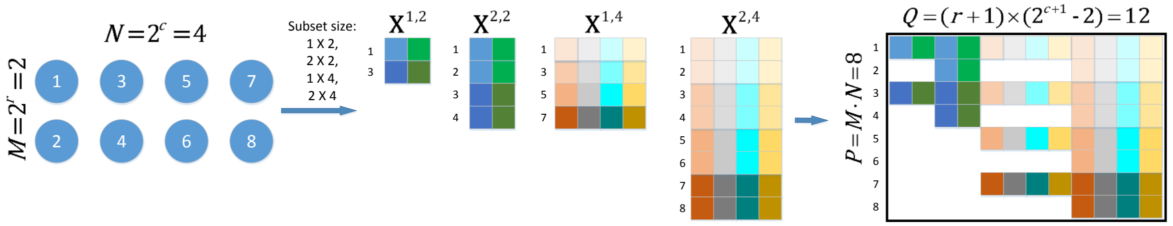
A. Impact of Antenna Subset Selection

For different subsets $m \times n$ of $M \times N$, we vary the number of rows (m), number of columns (n) and the arrangement ($m \times n$). Our objective is to understand how activating different subsets $m \times n$ affects the resultant beamwidths and directive gain.

(1) Increasing the number of rows (m) increases the directive gain in the desired beamforming direction while the beamwidth remains unchanged: It is natural that increasing m will increase the directive gain because of dependence on the total number of active elements (Fig. 5a). We also observe that the beamwidth in X-Y plane remains unchanged as adding additional row affects only the Y-Z plane beamwidth.

(2) Increasing the number of columns (n) increases the directive gain in the desired beamforming direction while decreasing the beamwidth: Since increasing n increases the number of elements in X-Y plane, more number of elements result in narrower beams (Fig. 5b). This means that changing the number of active columns will allow us to achieve different beamwidths.

(3) As the antenna element arrangement changes from square to rectangular to linear for the same number of antenna elements, the beamwidth decreases while the directive gain increases: Based on (1) and (2) above, it is clear that for the same number of antenna elements, the beamwidth decreases as their arrangement becomes more linear. These results are



(a) Example of a 2 × 4 antenna array. Each element has its own index

(b) Generating codebook for each subset. The activated element indexes are left to each matrix.

(c) Mapping and appending all codebooks to the final multi-level codebook. The rest of entries are assigned zero weight

Fig. 6: Procedure of codebook generation. In (b) and (c), each color represents one weight. Column-level color difference represents different patterns; row-level color density difference in each column represents different weights but belonging to the same pattern.

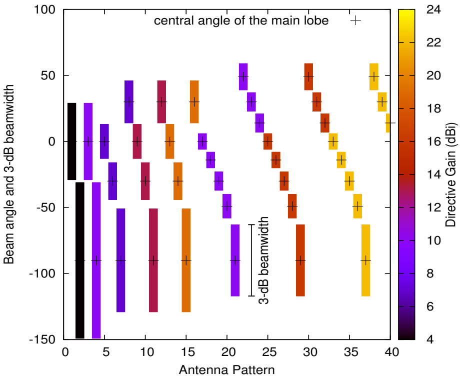


Fig. 7: Multi-level codebook for a 4 × 8 array. Proposed codebook covers different directions with many possible beamwidths

confirmed in Fig. 5c.

B. Multi-level Codebook Generation

As we observed above, varying number of antenna elements and their arrangement can achieve different beamwidths and gain for the desired beamforming direction. For a given $M \times N$ antenna array, different subsets of elements can be activated depending on beam pattern requirement.

A multi-level codebook is a $P \times Q$ matrix where $P = M \cdot N$ and Q is all possible beam patterns for various subsets of $M \times N$. Algorithm 1 provides the procedure for generating a multi-level codebook. Fig. 6 shows an entire procedure of the multi-level codebook generation for a 2×4 array. The algorithm first determines the subsets of elements for which their individual codebook can be developed. For each subset $m \times n$, we restrict the value of m and n to 2^x as they are known to maximize the directive gain [13] for phased arrays (Fig. 6a). It then calculates the phase shift weights using the DFT-based codebook design as mentioned in Equation (1). This DFT-based codebook design results in weights that are complex numbers. Also in each subset, we set the number of patterns to be generated as the number of columns and label each row using the indexes of antenna array elements (Fig. 6b). For those subsets which have multiple rows, such as $\mathbf{X}^{2,2}$ and $\mathbf{X}^{2,4}$, the algorithm duplicates the weights the first-row array elements have. Therefore in $\mathbf{X}^{2,2}$, index 2 and 4 have the same weights (same color and density) as index 1 and 3 in each pattern, respectively. This avoids the generation of beams with extremely narrow or extremely wide beamwidths that are often not useful in practice [14], [15]. The remaining elements outside the subset are assigned zero weight, and then all the entries of the codebook are combined into a multi-level codebook (Fig. 6c). As we select some column in the multi-level codebook, the algorithm is activating the corresponding array subset and beam. The directive gain corresponding to the multi-level codebook in Fig. 6c is depicted in Fig. 8. We see that different columns give different directive gain. For instance, the shape of directive gain in 1×2 $col1$ (the green column in $\mathbf{X}^{1,2}$) is the same as it in 2×2 $col1$ (the green column in $\mathbf{X}^{2,2}$), due to duplicated weights (colors) in index 2 and 4. However, the maximum gain of 2×2 $col1$ is higher, so we still regard these two patterns as different ones. Therefore, such a multi-level codebook provides a wide variety of directive gain and beamwidths.

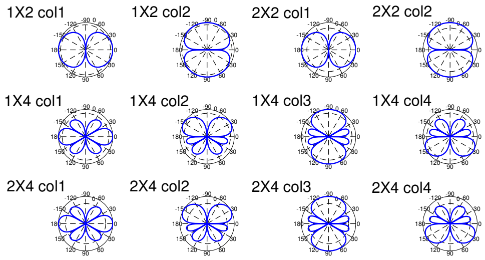


Fig. 8: Directive gain generated through multi-level codebook from Fig. 6. Since the number of columns Q is 12, the number of patterns is 12 as well.

Fig. 7 shows the results of the multi-level codebook generation procedure when applied to a 4×8 antenna array. For clarity, we only show patterns in one direction (90° to $-90°$). As we can see, a multi-level codebook results in many beams (patterns) that can cover different beamforming angles using a variety of beamwidths and directive gain. This way, depending on the client mobility, a wider or narrower beamwidth can be chosen with the appropriate gain value. Also, since beam adaptation in such multi-level codebook design is simply changing the weight vector, constant CSI or AoA/AoD monitoring is not required.

V. Sensor-Assisted Beamforming

Now we move our focus on how to use the multi-level codebook in the presence of mobility. In SAMBA, the AP leverages the client smartphone's accelerometer and magnetometer to predict its mobility and proactively adapt the beam. In this section, we describe how the sensor observations can be used in mobility prediction and how the AP performs beam adaptation using it.

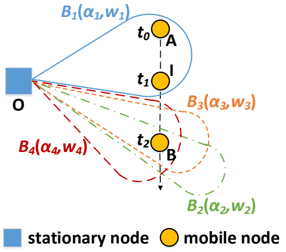


Fig. 9: Mobile client's heading direction. Its distance traveled and AP-client distance can be used to predict client's next location. For the next predicted point, the AP using multi-level codebook can have multiple beam choices (B_2, B_3 and B_4 above)

Consider a scenario presented in Fig. 9 where the mobile node (client) located at point A at time t_0 moves to point B at time t_2 via point I. We assume that at time t_0, the AP has already chosen a beam $B_1(\alpha_1, w_1)$ where α_1 is the beamforming angle and w_1 is the beamwidth from the codebook using the SLS procedure. Now, as the client moves to point B via point I, the AP has to switch to other beam by choosing one of the B_2, B_3 or B_4. The available choices depend on the output of the multi-level codebook designed in the last section. The objective is that the beam switching should happen in advance in order to maintain an uninterrupted connectivity.

A. Predicting Client Mobility

In order to proactively switch the beam without causing any disconnection, the AP should be able to *predict* the client's move to point B. We observe that such a prediction can be done using three pieces of information - (1) the distance between the AP and the client (d_{OI}) (2) the heading direction of the client and (3) the distance traveled by the client as it moves from point A to point I (or its speed). Using the three estimates, client's next location (point B) can be predicted.

(1) Estimating the AP-client distance: Different ways of estimating the AP-client distance have been proposed especially in the context of indoor localization [16]–[18]. The most widely used distance estimation method is based on the calculation of the round trip propagation time of the signal between the AP and the mobile node [18]. The technique has proven to be much more accurate when the estimation is done over direct LOS path between the endpoints. The challenge however is that in case of 2.4/5 GHz links, the LOS path might be obstructed by walls or other objects which makes the distance estimation much more difficult. Even in such cases, [16], [19] showed that Channel Impulse Response (CIR) can be used to identify the direct path. On the other hand, distance estimation in 60 GHz case is relatively simpler because majority of the times the communication happens only via the direct LOS path. Especially for the indoor use cases where SAMBA is applicable, LOS path is always available and can be used for the distance estimation. Hence, we use propagation time of the LOS path for the AP-client distance estimation in our case. The impact of blockage and NLOS path is also important, which is left as our future work.

(2) Estimating heading direction: The mobile node's heading direction can be derived using magnetometer sensor readings. Such a direction consists of three separate values (X, Y, Z). X is device's heading direction relative to magnetic north pole, Y is the direction relative to the ground where a value of 90^o means that the smartphone is placed in vertically standing position and Z shows the orientation change around the device's own axis (often referred as the "roll"). When a user is walking with a smartphone in her hand, the heading direction can be well estimated using the X value.

Note that when determining user's heading direction using magnetometer sensors, one critical aspect is to correctly recognize device's orientation. Because the estimation of heading direction would change depending on whether the user is holding the phone in her hand or the phone is kept in the pocket. This requires that phone's own orientation should be isolated (using Gyroscope sensors) from the heading direction to get an accurate estimate. We observe in our experiments that heading direction can be well estimated using the magnetometer sensors when the user holds the phone in her hand while walking (phone's heading is the same user's heading) as opposed to keeping the phone in pant's pocket. Because in our 60 GHz use cases such as streaming smartphone screen to TV set, the smartphone is typically carried in hand only, we do not apply any orientation correction on heading direction observations. Even if we consider the cases where smartphone is kept in other places such as pant's pocket while walking, the heading direction estimate can be corrected using the Gyroscope and Accelerometer based methods proposed in [16], [20].

(3) Estimating the distance traveled by the mobile node: In our case, since the user is walking inside the room with a smartphone in her hand, we can use the accelerometer readings to determine the walking speed, and then use it to find the estimated distance traveled by the user.

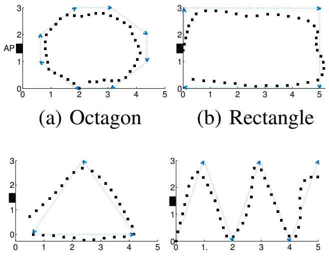
B. Using Mobility Prediction for Beam Adaptation

We now discuss how we can utilize mobility prediction and our multi-level codebook to perform beam adaptation. Algorithm 2 outlines the SAMBA procedure. The decision making in SAMBA is dependent on the observed SNR values from the client. It is assumed that the AP has already calculated the multi-level codebook using Algorithm 1. If the client's observed SNR is below the minimum threshold (SNR_{HT} - hard threshold), the AP considers the client to be disconnected and initiates a beam searching using SLS. Once the SLS procedure is complete, the AP knows a beam that it uses to connect to the client. Now as the client moves around, its observed SNR values vary. During this time, the mobile node also constantly monitors its accelerometer and magnetometer readings, and derives its heading direction and traveled distance as discussed in Section V-A. The client shares its heading direction and distance (or speed) readings with the AP.

In order to reduce the overhead of constantly providing sensor feedback to the AP, we use SNR_{ST} as a soft threshold. Whenever client's SNR is between SNR_{ST} and SNR_{HT}, then

Algorithm 2 SAMBA Procedure

```
Input:
    Multi-level Codebook $\mathcal{C}$, Hard SNR Threshold $\mathrm{SNR}_{HT}$,
    Soft SNR Threshold $\mathrm{SNR}_{ST}$
Procedure:
for time $t : 0 \rightarrow \infty$ do
    Get current SNR of the client ($\mathrm{SNR}_t$) under current beam $\mathrm{B}_t$
    if $\mathrm{SNR}_t$ is unknown then
        Initiate Sector Level Sweep (SLS)
    else if $\mathrm{SNR}_{HT} \leq \mathrm{SNR}_t < \mathrm{SNR}_{ST}$ then
        Receive (heading direction H, speed S) observations from the client
        Estimate the distance D to the client
        Use $\mathrm{H}, \mathrm{S}$ and D to get client's estimated location $\mathrm{L}_{t+1}$ for time $t+1$
        Let $\mathrm{SNR}_{B_t}^{L_{t+1}}$ be the expected SNR at location $L_{t+1}$ for the current beam $\mathrm{B}_t$
        if $\mathrm{SNR}_{B_t}^{L_{t+1}} > \mathrm{SNR}_{HT}$ then
            Continue using $\mathrm{B}_t$
        else
            For each $\mathrm{B}_i$ in candidate set of beams $\mathrm{C} \subset \mathcal{C}$, calculate $\mathrm{SNR}_{B_i}^{L_{t+1}}$
            Switch to the $\mathrm{B}_i$ with the minimum expected SNR but above $\mathrm{SNR}_{HT}$.
        end if
    end if
end for
```



(a) Octagon (b) Rectangle

(c) Triangle (d) Zigzag

Fig. 10: Sensor data (black dotted line) and ground truth (blue dotted arrow) of four pre-defined trajectories

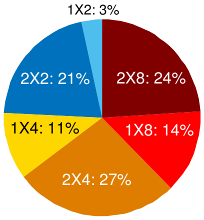


Fig. 11: Different beamwidths through subsets of 2 × 8 and their percentage usage in the zigzag trace

TABLE I: Percentage of connected time for real mobility traces

Trace shape	True location	Sensor-predicted location
Octagon	84.47%	89.38%
Rectangle	87.7%	89.33%
Triangle	85.83%	86.38%
Zigzag	71.42%	72.92%

only the client provides the sensor feedback to the AP. This avoids the unnecessary sensor reports to the AP when the client has reasonable SNR and/or is stationary. Now when the observed SNR is indeed between SNR_{ST} and SNR_{HT}, it indicates that the client is possibly near the boundary of the current beam, and it can possibly move out of the current beam. At this time, the client starts sending the sensor readings to the AP which triggers the mobility prediction on the AP. The AP uses client's heading direction and its traveled distance (or speed) along with its estimate of AP-client distance to predict client's location at the next time step (refer Algorithm 2).

Once the AP has calculated the client's projected location, it checks if the estimated SNR for the current beam for that location is sufficient or not. If the SNR is sufficient, the AP continues to use the current beam. If not, the AP prepares to switch to a different beam (changing beamwidth or beamforming direction) in anticipation of client's mobility. At this time, the AP scans its multi-level codebook and generates a set of candidate beams/patterns that can cover the client's predicted location. It then picks the beam with the minimum SNR value but above SNR_{HT} to guarantee that the beam with broader beamwidths is always chosen, and finally switches to that beam at the next time instance. Note that other criteria (e.g. the client's expected dwell time) can be applied to pick the best beam from the candidate set and we leave their evaluation to the future work.

If the client's mobility is not correctly predicted, the client's observed SNR becomes unknown after beam-switching. At this point, the AP is required to perform the beam searching. Note that considering the prediction methods of Section V-A, such cases of outages can happen when client's mobility is abrupt and does not follow a known pattern. We observe that such instances are unavoidable in the real-world indoor mobility traces which we discuss in the next section.

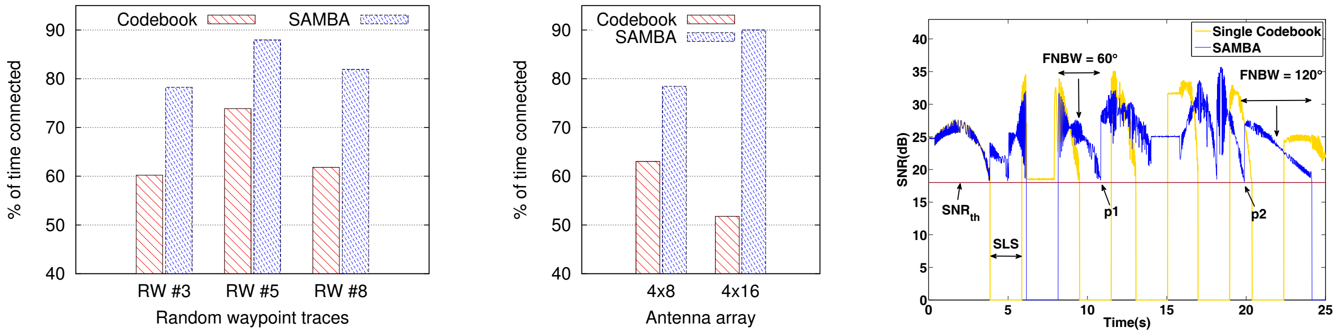
It is worth noting that we also consider the client to AP beamforming in SAMBA. The client to AP beamforming is similar to the beamforming from the AP to the client. In this case, the sensor observations can be used by the client itself for beamforming. Once the client has completed the SLS procedure, it has already established a beam towards the AP. Now, as the client moves around, it can use its heading direction, distance traveled and the AP-client distance to recalculate the estimated location of the AP and the beamforming direction. However, we do not consider the beam adaptation (client always uses the beam with the highest gain) in the client to AP case because the AP is stationary.

VI. Numerical Evaluation

Although there are a few 60 GHz chipset and antenna systems available, either they use a single horn antenna [21] or their programmability is not enough to implement SAMBA. The Presto platform proposed in [12] provides some control over switched beam functionality although it is not sufficient to perform real-time beam adaptation of SAMBA. Therefore, we rely on trace-driven simulations for evaluating SAMBA in which we first collect mobility/sensor traces using smartphone and then import them in MATLAB where the multi-level codebook is simulated. We also evaluate SAMBA using simulated mobility traces generated using the Random Waypoint mobility model. First, we discuss the evaluation using the real mobility traces and discuss the random waypoint traces after that.

A. Collection of Mobility Traces

We collect our sensor traces using two different Android smartphones - Google Nexus 5 and Samsung Galaxy S5.



(a) Performance of SAMBA under three random waypoint mobility traces

(b) Comparison of impact of different antenna array size on the performance of SAMBA

(c) SNR variation shows that SAMBA predicts the client mobility and utilizes larger beamwidths to maintain connectivity

Fig. 12: Performance comparison under Random Waypoint mobility traces

We use the “Andro Sensor” [22] app which can record accelerometer, magnetometer, heading direction and distance traveled (or speed of walking) at a sampling frequency of 20 Hz. We also create our own Android app that can calculate heading direction using the magnetometer readings. This app is used to cross-validate the accuracy of the Andro sensor data. We observe that Andro-sensor data closely matches the observations of our app.

As we discussed in Section V-A, we hold the phone in flat in our hand while collecting the sensor readings as opposed to keeping the phone in pocket. This orientation of the phone is most likely to be common across all 60 GHz use-cases such as streaming the smartphone screen to television or monitor. While collecting the traces, we use a room of size $3m \times 5m$, and assume a stationary position of the AP. Meanwhile, we record the ground truth of the traces by marking the room floor with ticks to measure distance. We also assume that the heading direction is well aligned with the ground truth trace at the starting point. By this way, only the local difference among sensor reading is taken care of rather than the actual degree reading. We walk on each pre-defined trajectory multiple times back and forth for all four traces. The traces of sensor reading and corresponding ground truth are depicted in Fig. 10. Due to the measurement error of magnetometer and gyroscope sensors, the ground truth traces are slightly different from the traces we get from the sensors. Note that indoor mobility of users might be better represented using random traces, we select these mobility traces and trajectory shapes to evaluate the error in sensor-based prediction in few representative cases. We will evaluate the random mobility cases in Section VI-C.

B. Performance of SAMBA on Real Traces

We use 2×8 as the given antenna array and generate the multi-level codebook. Also, 18 dB is used as the value of hard SNR threshold SNR_{HT}. As per the standards specifications [1], [5], 18 dB SNR is sufficient for many MCS levels that can provide a throughput of over 2 Gbps.

Table. I shows the percentage of the time the client is connected to the AP. We evaluate the percentage for two locations - device’s true location and device’s location as predicted using sensor observations. For sensor-predicted location, we use the sensor data both to generate beams and to calculate the target location in each time stamp; that is to say, we suppose that sensor data was perfectly correct and there was no measurement error for locating the target during the connection between the AP and the target. For true location, we still use the beam generated from sensor data but calculate the target location based on the ground truth trace in each corresponding time stamp.

We observe that when turning angle is smaller at each turn (octagon $\gtrsim$ rectangle $>$ triangle $>$ zigzag), our target location prediction performs better which in turn results in improved coverage of target. Note that the beam in each step emitted from the AP might not perfectly point to the actual target’s trajectory. As we can see, it is natural that the percentage values for the true location case are slightly lower than the sensor-predicted location cases, however, the percentage value difference is less than 5% for all four traces. This lower difference indicates that the error in prediction of true location using sensor measurements is relatively low. The highest difference ($\approx 5\%$) between the two is observed for the octagon trace. This can be attributed to smaller turns in the trace compared to other traces which leads to more error in target location prediction, because the sensors might not be quite sensitive to small angle changes. The zigzag case has the worst connectivity due to 1) many *sharp* ‘V’ turns, which require frequent SLS to find the target; 2) farther distance from the AP requires the use of narrower beams that are more susceptible to target mobility. Fig. 11 shows the fraction of the time different beamwidths are utilized for the zigzag mobility trace. We observe that the subsets of larger beamwidths with adequate SNR (e.g. 2×2, 2×4) are chosen majority of the times. The other subsets providing narrower beamwidths (1×2, 1×4 and 1×8) are relatively less utilized. This means that SAMBA performs the beam dilation in the presence of mobility to avoid outages.

C. Random Waypoint Mobility

Now we evaluate SAMBA’s performance using the mobility traces generated with the random waypoint model. The random waypoint mobility model has been proposed for simulating in-room mobility in many use cases (e.g. conference room) suggested by [1] standard. We consider a larger room size of $20m \times 20m$ where we set the client speed to closely resemble user’s walking speed which is known to vary between 1 to 2 m/s [23]. We use the array size of 4×8 which is necessary to achieve a sufficient gain to cover a large room size. We compare the performance of SAMBA with the performance of single conventional codebook scheme. In the conventional codebook scheme in which no subset of 4×8 is considered, roughly only one narrow beamwidth is available for any beamforming angle, and there is no mobility prediction using the sensor data. The results of percentage connected time for three random mobility traces are shown in Fig. 12a and the average performance across 10 traces is shown in Fig. 12b (4×8 case). We can observe that SAMBA achieves the percentage connection time of around 80% (average of 10 simulated traces) significantly improving the performance compared to the single codebook case. This is because it has to frequently perform SLS due to high mobility outages outside its narrow beams in the absence of multi-level codebook and sensor-based prediction.

We repeat the evaluation using simulated traces for another size of antenna array - 4×16. The results are shown in Fig. 12b. As expected, due to the availability of more elements, number of possible beamwidths in multi-level codebook increases which results in overall improvement in performance of SAMBA in the 4×16 case compared to the 4×8 one. However, it is interesting to observe that conventional codebook performs even worse (% connection time of around 52%) with 4×16. This is because in presence of more elements, single codebook results in increasingly narrower beamwidths. Such narrower beamwidths in the absence of sensor-based prediction fails to deal with client mobility, resulting in very high number of beam searching instances.

Fig. 12c shows the SNR variation (for Trace RW#8) for SAMBA and single codebook cases. As we can see, SAMBA predicts the mobility of the client at point $p1$ and $p2$, and proactively changes the beam in order to avoid disconnection. Also, with the use of our novel multi-level codebook, it often adapts to larger beamwidths (FNBW[1] of 60^o and 120^o in Fig. 12c) which provide substantial improvement in % connection time as compared to conventional codebook.

VII. Conclusions

In this work, we proposed a sensor-assisted beam adaptation technique (SAMBA) to address the mobility challenge in 60 GHz network. We showed that when antenna selection is incorporated in the codebook design, it is possible to generate a multi-level codebook that can provide richer choices of beamwidths and directive gain in different beamforming directions. We then show that mobile device's accelerometer and magnetometer sensors can be used to better predict its mobility, and the prediction combined with multi-level codebook can significantly improve the overall connection time. We evaluated our scheme with real mobility traces. We plan to extend and evaluate SAMBA using software radios or programmable 60 GHz hardware depending on their availability in the future. We are also investigating interference mitigation schemes for dense indoor deployment of 60 GHz links.

[1] First Null Beam Width is the angular difference between the first nulls around the main lobe

References

[1] IEEE P802.11adTM/D4.0, "Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 3: Enhancements for very high throughput in the 60 ghz band, ieee computer society," *IEEE Computer Society*, July 2012.

[2] WiGig Alliance. [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wigig-certified

[3] S. Shankar N., D. Dash, H. El Madi, and G. Gopalakrishnan, "WiGig and IEEE 802.11ad - For multi-gigabyte-per-second WPAN and WLAN," *ArXiv e-prints*, Nov. 2012.

[4] A. Technologies. Wireless LAN at 60 GHz - IEEE 802.11ad explained. [Online]. Available: http://cp.literature.agilent.com/litweb/pdf/5990-9697EN.pdf

[5] IEEE Std 802.15.3cTM-2009, "Part 15.3: Wireless medium access control (mac) and physical layer (phy) specifications for high rate wireless personal area networks (wpans) amendment 2: Millimeterwave-based alternative physical layer extension," *IEEE Computer Society*, October 2009.

[6] "Wilocity 802.11ad Chipset." [Online]. Available: http://wilocity.com/resources/Wil6200-Brief.pdf

[7] S.-K. Yong, P. Xia, and A. Valdes-Garcia, *60GHz Technology for Gbps WLAN and WPAN: from Theory to Practice*. John Wiley & Sons, 2011.

[8] N. Celik, M. Iskander, R. Emrick, S. Franson, and J. Holmes, "Implementation and experimental verification of a smart antenna system operating at 60 ghz band," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 9, pp. 2790–2800, Sept 2008.

[9] K. Ramachandran, N. Prasad, K. Hosoya, K. Maruhashi, and S. Rangarajan, "Adaptive beamforming for 60 ghz radios: Challenges and preliminary solutions," in *Proceedings of the 2010 ACM International Workshop on mmWave Communications: From Circuits to Networks*, ser. mmCom '10.

[10] J. Wang, Z. Lan, C. woo Pyo, T. Baykas, C.-S. Sum, M. Rahman, J. Gao, R. Funada, F. Kojima, H. Harada, and S. Kato, "Beam codebook based beamforming protocol for multi-gbps millimeter-wave wpan systems," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 8, pp. 1390–1399, October 2009.

[11] K. Amiri, D. Shamsi, B. Aazhang, and J. Cavallaro, "Adaptive codebook for beamforming in limited feedback mimo systems," in *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, March 2008, pp. 994–998.

[12] X. Tie, K. Ramachandran, and R. Mahindra, "On 60 ghz wireless link performance in indoor environments," in *Proceedings of the 13th International Conference on Passive and Active Measurement*, ser. PAM'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 147–157. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28537-0_15

[13] A. Valdes-Garcia, S. Nicolson, J.-W. Lai, A. Natarajan, P.-Y. Chen, S. Reynolds, J.-H. Zhan, D. Kam, D. Liu, and B. Floyd, "A fully integrated 16-element phased-array transmitter in sige bicmos for 60-ghz communications," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 12, pp. 2757–2773, Dec 2010.

[14] L. Chen, Y. Yang, X. Chen, and W. Wang, "Multi-stage beamforming codebook for 60ghz wpan," in *CHINACOM, 2011 6th International ICST Conference on*, Aug 2011, pp. 361–365.

[15] L. Zhou and Y. Ohashi, "Efficient codebook-based mimo beamforming for millimeter-wave wlans," in *PIMRC 2012*.

[16] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," ser. ACM MobiSys '14.

[17] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," ser. ACM MobiSys '13.

[18] D. Giustiniano and S. Mangold, "Caesar: Carrier sense-based ranging in off-the-shelf 802.11 wireless lan," ser. ACM CoNEXT '11.

[19] Z. Zhou, Z. Yang, C. Wu, W. Sun, and Y. Liu, "Lifi: Line-of-sight identification with wifi," in *INFOCOM, 2014*.

[20] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," ser. ACM MobiSys '14.

[21] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall, "Augmenting data center networks with multi-gigabit wireless links," ser. ACM SIGCOMM '11.

[22] "Andro Sensor." [Online]. Available: https://play.google.com/store/apps/details?id=com.fivasim.androsensor

[23] C. Willen, K. Lehmann, and K. Sunnerhagen, "Walking speed indoors and outdoors in healthy persons and in persons with late effects of polio," *Journal of Neurology Research*, vol. 3, no. 2, pp. 62–67, 2013.