Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support

Teng Wei
University of Wisconsin - Madison
twei7@wisc.edu

Xinyu Zhang
University of California San Diego
xyzhang@ucsd.edu

Figure 1: Multi-AP networks: APs are partially overlapped and connected to a backend server via wireless backhaul.

mostly been focusing on point-to-point, stationary links. To enable seamless coverage and mobility support, 60 GHz networks need to overcome fundamental barriers that do not exist in the prior low-frequency counterparts. To compensate for the intrinsic attenuation losses due to short signal wavelengths, 60 GHz radios adopt phased-array antennas which comprise many planar patch elements to form directional beam patterns. Although these beams are electronically steerable, their joint coverage is still limited due to the inherent half-space coverage of patch antennas [10]. So the phased-array bears a limited field-of-view (FoV), much like a camera. And the signal strength largely depends on whether the receiver falls in the transmitter’s FoV. Consequently, achieving stable 60 GHz connectivity, even at room-level, becomes a nontrivial task.

In this paper, we propose Pia, a robust 60 GHz network architecture that can provide room-scale coverage at multi-Gbps bit-rate. Pia is tailored for emerging applications such as wireless virtual-reality [2], augmented-reality [11], uncompressed miracast [4], all requiring Gbps connectivity between a mobile device and a backend server for computational offloading or graphical rendering. Our basic idea is to deploy multiple cooperating APs (e.g., at the corners of the room), each covering a “picocell” region that can complement others’ blind spots. No matter how the client device moves and rotates, it is likely to fall in the FoV of at least one of the APs (Fig. 1). The APs can be 802.11ad-compatible access points or dedicated relaying devices [2], which connect to a backend server through Ethernet cables or fixed-beam wireless backhaul [1].

The idea of multi-AP coordination has been investigated extensively in legacy WiFi or cellular networks [12–16], and the key problem is to determine which AP to connect to, based on signal strength, traffic load, etc. But the new characteristics of 60 GHz bring unique dimensions to the problem. Since each 60 GHz AP may possess hundreds of beam directions, a straightforward way of selecting the AP will entail trial-and-error probing across all these directions for all APs. When multiple clients coexist, multiple iterations of probing are needed to negotiate the best spatial reuse, which compounds the overhead. Most critically, the probing needs to be done not only when a client changes its location, but also its orientation. Consequently, the probing overhead may overwhelm normal data transmission.
Pia overcomes this issue by leveraging the 5-DoF pose information (x, y, z position and polar/azimuth orientation) that is available on many mobile devices (e.g., VR headsets [17] and smartphones/tablets with visual-inertial sensing capabilities [18,19]). More specifically, Pia employs a model-driven approach to predict the best AP based on a client’s pose, and to assign the best beam pattern each AP should use to maximize the spatial reuse among multiple clients. To this end, we identify 4 unique design challenges.

First, our measurements indicate that the 60 GHz throughput often experiences a catastrophic drop as a client moves out of the AP’s FoV, so the conventional multi-AP protocols that react after throughput change can no longer sustain robust connectivity. We thus design a pose-assisted link predictor that allows the client to proactively switch to a new AP, before the current link quality drops to an intolerable level. This decision builds on a prediction of the client’s short-term pose change, paired with a simple model of the AP’s coverage which does not require probing all beam patterns.

Second, a vast literature in directional-antenna networking at low-frequency bands assumed cone-shaped beams [20]. In contrast, practical 60 GHz phased-arrays have imperfect directional beam patterns that often comprise multiple sidelobes, causing irregular interference patterns across spatial angles. Pia addresses this challenge using a pose-assisted spatial sharing mechanism, which jointly optimizes the AP selection and beam assignment, to maximize the concurrent transmission opportunity when multiple clients coexist.

Third, the above two mechanisms assume Pia knows the AP’s relative position and orientation within the client’s coordinate. However, the actual APs are often placed in an ad-hoc manner. To meet this challenge, we design a statistical algorithm, called AP-pose sensing (APS), that can estimate the AP’s pose based on a random set of link-quality measurement at the client side.

Finally, Pia’s interference management mainly models the line-of-sight (LOS) link quality. But the model can be occasionally disturbed by non-line-of-sight (NLOS) reflections from close-by objects (e.g., concrete walls or metal cabinet). We design a novel compressive angle estimation method, that fuses the pose information with the link quality measurement, so as to discriminate the reflection paths and model their impacts separately.

The contributions of Pia can be summarized as follows.

(i) We propose to use pose information as a fundamental primitive, and identify the associated challenges/opportunities to facilitate robust connectivity within a 60 GHz multi-AP architecture.

(ii) We design pose-assisted link prediction, spatial sharing, and AP-pose sensing mechanisms, to enable efficient link interference management for practical 60 GHz clients.

(iii) We implement Pia on a COTS testbed with 4 60 GHz radios each with a 4×8 phased array. Our experiments show that Pia’s pose-assisted AP switching effectively reduces the hazard times (catastrophic throughput drops) by 4.2× to 6.3×, and the spatial sharing enhances the network capacity by 76% compared to the 802.11ad.

2. BACKGROUND AND OVERVIEW

2.1 Limited FoV of the 60 GHz Radio

Impact of FoV alignment on link quality. Early studies of 60 GHz phased-array [10] indicated that its coverage is often limited to less than half-space. To elucidate the problem, we set up two 802.11ad laptops, as AP and client, respectively; both equipped with the Qualcomm QCA6300 series chipset [21]. We measure the link throughput when rotating the client in front of the AP which is 1.5 m away and facing to the client directly (more details of the setup are in Sec. 5). From the results (Fig. 2(a)), we can make two observations. First, the client’s high-throughput coverage area forms an angular sector (around 170°), which we refer to as its FoV. Once the AP falls outside the client’s FoV, the throughput drops sharply, from 2.2 Gbps to several hundred Mbps. Note that the antenna’s FoV is different from its beam pattern. Each beam pattern is much narrower than the FoV itself (more in Sec. 4.1). Second, there is no ‘notch’ inside the FoV—all the beam patterns together can fully cover the FoV, ensuring a consistently high throughput.

The limited FoV practically exists in all 60 GHz devices, because the phased array’s front-side comprises many planar antenna elements, each having a FoV of 80° to 180° [10,22]. The backside has very weak signal emissions, because it is grounded by a metal plane, and often faces inward the host device which causes strong attenuation. Following this common practice, 802.11ad-compatible laptops (e.g., Acer P446 and Dell Latitude E7240) typically install the phased array close to the outer surface of the lid. The TP-CAST [8] 60 GHz adapter is mounted atop a VR headset, with the phased array pointing towards the ceiling. For smartphones, antennas are recommended to be placed on the top or bottom surface [23].

Note that, besides the host device itself, the human user can further block the phased array. In this work, we assume the user of the device always falls outside the phased array’s FoV, i.e., blocking the backside only. This is a valid assumption considering the antenna placement principle [24] (in the most unobstructed zone during regular use), and also the fact that the user moves synchronously with the device in practical applications, e.g., mobile gaming and wireless VR. Blockage of other users can be alleviated by careful AP deployment/planning, which will be discussed in Sec. 7.

Fig. 2 (b) also plots the measured throughput as link distance increases, with client and AP falling in each other’s FoV. We find the link can sustain a high throughput over a reasonably long distance. Even at 13 m, the throughput remains at 1.8 Gbps, much higher than the out-of-FoV case. This range is consistent with previous measurements [25, 26]. It indicates that the devices’ orientation, which determines FoV alignment, plays a more critical role than link distance in maintaining robust room-level connectivity.

Profiling the beam-steering under mobility. Can the beam steering on commercial 802.11ad radios be fast enough to adapt to the user movement? To answer this question, we walk inside the network area, holding the client device with natural pose variations, but keeping it within the FoV to an AP. Fig. 3(a) further plots the throughput measurement over time. Although the throughput shows higher variations, the mobile in-FoV case can still main-
tain 50% at worst and 80% on average compared to the stationary case, and 3~5x higher than the out-of-FoV case. The beam index changes rapidly in the mobile case (Fig. 3 (b)), indicating the real-time beam adaption of 802.11ad can accommodate user mobility with reasonable efficiency, at least at walking speed. So the in or out of FoV relation becomes the dominant factor that determines link throughput.

2.2 Pia Operations

To facilitate seamless coverage and mobility support for 60 GHz networks, Pia adopts a two-stage workflow (Fig. 4). The sensing stage takes the 60 GHz link status information, i.e., modulation and coding scheme (MCS) and beam index, along with the pose information from the mobile client as input, and runs the AP-pose sensing (APS) algorithm (Sec. 3.2) to estimate the APs’ pose within the client’s coordinate system. With the same input, Pia also runs a compressive angle estimation algorithm to estimate the location/orientation of major reflectors that may affect the network performance (Sec. 4.2). Both schemes are one-time initialization procedures that run before putting the network into use. They need to be repeated only when the APs are redeployed or the environment changes significantly.

During the running stage, each client periodically feeds back its pose information to the backend server, using its current AP as a relay. The server runs the link prediction (Sec. 3.1) and interference management (Sec. 4) algorithms, and returns the decisions to the APs/clients. Pia’s decision-making mechanisms are proactive in nature, based on a prediction of the long-term (> 500 ms) pose (Sec. 3.1.2). The pose sampling period and feedback latency are relatively negligible compared with the look-ahead time.

3. Pia DESIGN

3.1 AP Selection

We first consider a single mobile client and focus on how to dynamically select the AP to maintain high link throughput. This issue resembles the classical AP handoff problem in WiFi [12–16]. A common solution is to probe each AP and select the one with highest potential throughput. However, such approaches are reactive, and more critically, the many-beam phased-arrays in 60 GHz networks will cause huge probing overhead. For example, to probe all possible beams, an 802.11ad client needs to wait for the beam training beacon, which occurs only once per beacon period (typically 100 ms) [27]. By the time a proper AP is identified, the link may have already suffered from a long period of outage. Pia adopts a different design principle. It predicts the best AP based on the client’s pose, without explicit probing. It proactively decides on which AP the client should switch to, and when. The AP/beam selection algorithms will execute each time when a new pose data arrives. Pia will inform AP the new beam selection only when algorithm’s output changes. We now describe the mechanisms in detail.

3.1.1 Pose-Assisted Link Predictor

In Pia, each pose sample, denoted as $P = [x, y, z, \theta, \phi]$, represents the 3D location, polar and azimuth angle of the norm of the phased array in a spherical coordinate system (Fig. 5). We can ignore the rotation angle around the antenna’s norm direction because such rotation does not change the FoV alignment relation between the client and AP.

Based on the empirical insights in Sec. 2.1, each client discriminates the APs using a binary metric 0/1, denoting whether an AP is estimated to be within or out of FoV. It should be noted that this binary metric is only applied for AP selection purposes. Finer-grained metrics will be used for multi-client interference management (Sec. 4.1). In addition, for picocell coverage within a room area, the power budget of commodity 60 GHz devices already provides consistently high throughput for the in-FoV case (Sec. 2.1). For larger scale networks such as outdoor small-cells, the AP-to-client distance may have a non-negligible impact, but can be easily modeled in Pia given the location information.

The predictor takes the AP’s and client’s poses as input. It predicts the link status as 1 if they mutually fall within each other’s FoV (Fig. 6), and 0 otherwise. We define the FoV angle as from AP to client, following 3D geometry:

$$\text{Link State} = \begin{cases} 1, & \phi_s \leq \beta_s \text{ and } \phi_c \leq \beta_c \\ 0, & \text{Otherwise} \end{cases}$$

Fig. 7 plots an example trace of the link predictor, as the client walks within a 2×3 m² region with natural orientation changes (More setup details are available in Sec. 6). To obtain the ground-truth link state, we measure the link throughput corresponding to each pose sample, and then convert it into 1/0, using a threshold of 1.6 Gbps that can reliably distinguish the in-FoV and out-FoV (Sec. 2.1). We highlight the false in-FoV predictions (i.e., out-FoV predicted as in-FoV) which mislead the client to switch to a low-throughput AP. We find the prediction error is only 9.11% across all
the 1324 pose samples. More importantly, the errors mostly concentrate on the boundary area i.e., when transitioning from in/out-FoV to out/in-FoV. We will show that such errors have marginal impacts because Pia reacts well before the transition occurs (Sec. 3.1.3).

3.1.2 Pose Predictor

Pia needs to estimate the link status for the near future based on a prediction of the client’s pose. For simplicity, it adopts a classical kinematic model, Continuous White Noise Oscillation (CWNA) [30], for pose prediction, although other kinematic models can be applied as well. CWNA assumes zero acceleration, i.e., both the linear and angular velocities are stable within a very short duration. Thus, the pose \( \hat{P}_e(t + 1) \) at time \( t + 1 \) can be predicted as:

\[
\hat{P}_e(t + 1) = P_e(t) + \Delta \hat{P}_e(t),
\]

where \( \Delta P_e(t) = P_e(t + 1) - P_e(t) \) is the pose velocity.

A key question here is how far should Pia predict ahead of time. Let \( L \) denotes the look-ahead time. \( L \) should be long enough to ensure the AP switching can finish timely, yet not too long to make the prediction unreliable. In our Pia implementation (Sec. 5), the AP switching time is negligible (\(< 10 \) ms), so we empirically set \( L \) to 500 ms, roughly the scale where human movement velocity remains coherent [30].

3.1.3 AP Switcher

To avoid the adversarial impact of false in-FoV predictions (Sec. 3.1.1), Pia leverages the transition area where different APs’ FoVs partially overlap (Fig. 8 (a)). It makes AP switching decision ahead of time before the link status of current AP degrades, so as to avoid the boundary region vulnerable to false prediction. To realize seamless switching, the APs should be deployed with partially overlapping FoVs. However, we will empirically verify that, even when this requirement is not satisfied, Pia’s AP switcher can still minimize the client’s outage duration (Sec. 6.2).

To determine the desired AP and appropriate switching time, Pia’s AP switcher takes the predicted link status for all APs from time \( t + 1 \) to \( t + L \) as input. Fig. 8(b) shows an example time series of link status (0/1) prediction for two APs, computed based on the pose data of headset and AP using Eq. (4). Such information forms a table-like data structure. Pia then makes the switching decision following three basic rules: (i) Prepare a switching only if the current AP has at least one 0 value in the table. Otherwise, the AP will remain in the client’s FoV in the look-ahead duration \( L \) and switching is unnecessary. (ii) Look for other APs that have overlapped 1s with current AP in the predicted time span, and select the AP with smallest AP-to-client distance as the AP to switch to. This essentially enables Pia to react within the transition area, and choose an alternative AP with potentially strongest RSS. (iii) Choose the center of overlapped 1s as the switching time, so as to avoid the boundary regions vulnerable to prediction errors (Fig. 8).

There are two exceptional cases that Pia needs to handle in order to minimize the impact of potential link outage: (i) When no other APs have any 1s, Pia needs to stick to the current AP. (ii) When no overlapped 1s exist, Pia will switch to the AP that has the earliest 1 in the predicted time span.

3.2 AP-Pose Sensing

The above link predictor assumes the APs’ poses are known. Ideally, a user can measure APs’ poses during deployment. However, in practice, the phased array antenna is commonly sealed inside the device, and the exact direction where its FoV points to is invisible to the end user. Pia’s APS algorithm overcomes this barrier by automatically estimating each AP’s pose, which will account for the antenna array placement inside the device. Although certain localization schemes [31] may help obtain the AP’s \((x_s, y_s, z_s)\) coordinate, they require at least three reference devices. Pia’s use cases typically do not satisfy this requirement because only one reference device, i.e., the mobile client, exists with known position. Besides, these schemes need access to fine-grained PHY-layer information such as phase, and often require precise carrier clock calibration between the transmitter and receiver, which is not viable on commodity 60 GHz hardware. To enable 5-DoF AP-pose sensing for a wider range of 60 GHz devices, we devise a statistical algorithm that only needs the link throughput measurement as input.

Collecting the sensing data. APS is called during the sensing stage of Pia, when the user randomly walks within the network coverage. Meanwhile, the client device records its pose and link throughput with respect to a connected AP. Then, Pia converts the link throughput into binary link status in the same way as the experiments in Fig. 7. Since a client device can estimate the link throughput for multiple APs simultaneously, a user only needs to collect the sensing data for all APs with a single walk. Besides, the user’s walking trace needs not cover every spot in the area.

Statistical estimation of the AP pose. Let \( S_0 \) denote a set that contains all legitimate AP’s pose values, and \( S_1 \) be the pose samples in the sensing trajectory. Then the optimal estimation of the AP pose should result in the minimum error in matching the measured in-FoV/out-FoV samples. We formulate this optimal estimation \( \hat{P}^* \) as one that minimizes the weighted false in-FoV errors \((w_{FI})\) and false out-FoV errors \((w_{FO})\) for a given set of samples:
Algorithm 1 AP-Pose Sensing

\begin{algorithmic}
1: \textbf{procedure} $\text{APPoseSensing}(S_t, T_t, w_{\text{tag}}, w_{\Psi})$
2: \hspace{1em} $[x_d, y_d, z_d] = \text{ellipsoidCenter}(S_t)$ \hfill $\triangleright$ Find the sweet spot
3: \hspace{1em} $D = APLocList(S_t)$ \hfill $\triangleright$ Generate list of legitimate AP locations
4: \hspace{1em} $v_{\text{Min}} \leftarrow \text{inf}$, $P^* \leftarrow \text{NULL}$ \hfill $\triangleright$ Initialize variables
5: \hspace{1em} for $\{x_s, y_s, z_s\} \in D$ do
6: \hspace{2em} $\theta_s, \varphi_s = \text{toAngle}([x_s, y_s, z_s], [x_d, y_d, z_d])$ \hfill $\triangleright$ To orientation
7: \hspace{2em} $P_s \leftarrow [x_s, y_s, z_s, \theta_s, \varphi_s]$, $v_s \leftarrow 0$ \hfill $\triangleright$ Candidate AP pose
8: \hspace{1em} end for
9: \hspace{1em} for $P \in S_t$, $t_s \in T_t$ do
10: \hspace{2em} $v = v + \text{inFoV}(P_s, P_j) \ast (1 - t_j) \ast w_{\Psi}$
11: \hspace{2em} \hfill $+(1 - \text{inFoV}(P_s, P_j)) \ast t_j \ast w_{\Psi}$ \hfill $\triangleright$ Eq. (5)
12: \hspace{2em} end for
13: \hspace{1em} if $v < v_{\text{Min}}$ then \hfill $\triangleright$ Minimize weighted error
14: \hspace{2em} $v_{\text{Min}} \leftarrow v$, $P^* \leftarrow P$
15: \hspace{1em} end if
16: \hspace{1em} end for
17: \textbf{end procedure} \hfill $\triangleright$ Return the estimated AP’s pose
\end{algorithmic}

\[ P^* = \arg \min_{P_s \in S_s} w_{\Psi} \sum_{j=1}^{|S_s|} \text{inFoV}(P_s, P_j) \ast (1 - t_j) + w_{\Psi} \sum_{j=1}^{|S_s|} (1 - \text{inFoV}(P_s, P_j)) \ast t_j, \]

where $S_t$ represents the number of samples in $S_s$, and $t_j$ is the ground-truth link status (0/1) value. Function $\text{inFoV}(\cdot)$ determines if the client and AP are within each other’s FoV (Eq. 4) and returns a binary (0/1) indicator. $P_s$ denotes the AP’s pose and is the only variable we need to solve. We choose to minimize the weighted error because the false in-FoV error is more harmful to the link stability than the false out-FoV error (Sec. 3.1). Thus, it deserves a larger weight in the optimization ($w_{\Psi} : w_{\Psi} = 4:1$ in Pia).

However, Eq. (5) is non-convex, and solving it directly will be computationally expensive due to the large space of $S_s$. Suppose we partition locations in the unit of 10 cm and rotations in $3^\circ$. Then the number of candidate poses for a typical-size room ($120 \text{ by } 120 \text{ by } 10 \text{ m}^3$) will be $2^{28}$.

To prune the search space, we leverage a sweet spot where the axial direction of the AP’s FoV passes through. At this location spot, the in-FoV estimation should have a minimal error. To locate such a spot, consider an example trace of sensing data in Fig. 9, where we mark all locations with “in-FoV” link status as red dots. Using the approach in [32], Pia finds a 3D ellipsoid of minimum volume covering all the red dots. The ellipsoid’s center is identified as the sweet spot because steering the AP’s FoV to it will maximize the number of red dots the AP covers. Afterward, Pia searches over the 3D space using Eq. (5) to determine the AP’s location. The search space becomes much smaller because it only involves 3-DoF. Since the sweet spot uniquely specifies the axial direction of the AP’s FoV, the AP’s orientation can be directly determined for a given location without searching. Algorithm 1 summarizes the above procedure of APS. Moreover, Pia can detect the AP position change after initial sensing, because an inaccurate AP pose will significantly reduce the AP selection accuracy (Sec. 6.1). When the AP selection reliability degrades below a certain threshold, Pia will notify the user to recalibrate the AP’s pose.

4. POSE-ASSISTED INTERFERENCE MANAGEMENT

Despite the 7 GHz of spectrum for 802.11ad, there only exist 4 orthogonal channels. Multiple clients sharing the same channel can become inevitable in dense networks, and when the AP-to-server wireless backhaul needs dedicated channels. The 802.11ad standard [5] specifies a spatial sharing mechanism, where directional links can sense their mutual interference. Concurrent transmissions are allowed only among interference-free directional links. However, to allow other links to sense its interference, each link needs to first transmit in an exclusive service period [5]. This repeats whenever its beam changes, which ultimately reduces spatial reuse, especially when there are multiple mobile links. We now describe how Pia overcomes the challenge.

4.1 Improving the Spatial Sharing Opportunity

4.1.1 Beam Strength Vector

Pia uses beam strength vector (BSV) as the core data structure to arbitrate the AP beam assignment for multi-client scenarios. BSV is defined as a vector characterizing the RSS values (w.r.t. the AP) across different transmit beam patterns. An example BSV is shown in Fig. 10 (a), measured using our 60 GHz AP and a customized software radio (Sec. 5) separated by 2 m. To obtain the BSV in commodity 60 GHz devices, there are two approaches:

**Measurement from radio.** First, one can leverage 802.11ad’s built-in beam training protocol. At the beginning of each beacon interval (BI), the AP transmits multiple beacon frames, each using a different beam pattern. The client can record the RSS of each transmit beam pattern, which forms the BSV. This approach doesn’t rely on the pose information and can be applied to general scenarios.

**Prediction from pose.** However, an AP sends out beacon frames at a fixed time interval, e.g., 100 ms, which is too long for delay-sensitive VR applications. Instead of waiting until the beginning of a BI, Pia chooses to leverage the pose information to directly derive the BSV. It first calculates the polar/azimuth angle looking from the AP’s pose to the client’s position (similar to Eq. (2) and (3)). It then looks up the gains of all beam patterns in the view-angle direction. The angular beam pattern can be obtained from the phased-array datasheet or using a one-time measurement (Sec. 5). Fig. 10 (b) illustrates a 2D example of this procedure, where the left plots the gains of different beam pattern indexes at different angles, and the right shows the corresponding BSV at angle $-36^\circ$.

4.1.2 Pose-Assisted Spatial Sharing

To enhance the spatial reuse opportunity, we introduce a novel interference management framework, which leverages the pose information to simultaneously determine the best AP and beam pattern for each client. Our key insight is that practical phased-arrays are not perfectly directional. Due to the use of discretized beam-forming weights (i.e., codebook entries) [10], each beam pattern may bear a main lobe along with multiple weaker side lobes. Fig 11 illustrates example beam patterns of our AP device, measured using a software radio (Sec. 5). Therefore, instead of greedily choosing the beam with strongest main lobe for each link and then measure whether the links can coexist without interference as specified by 802.11ad [5], Pia can judiciously assign beams with weaker lobes, which may sacrifice the RSS of certain links but create more spatial reuse opportunities for them.

We formulate this basic idea as a joint AP selection and beam selection problem. For clarity, we will focus on the downlink and optimize the AP’s beam selection because the downlink traffic will be the dominance in the VR application. However, the extension to uplink and client-side optimization can follow the same principle since the client’s pose is available. Our objective is to find the optimal AP and beam assignment that maximizes the overall link quality using beam strength map (BSM), a data structure that aggregates the BSVs between each pair of AP and client. Each row of BSM is a BSV for a client and an AP. Fig. 12 plots an example mea-
measured using our 3-AP 3-client testbed (Sec. 5). To account for the signal attenuation over distance, Pia subtracts the predicted BSV by the pathloss of Frii’s model [33] which takes the AP-to-client distance as input. Note that Pia does not need to predict client’s absolute RSS because the spatial sharing mechanism only relies on the ratio between BSVs.

Suppose there are \( N_c \) clients and \( N_A \) APs. Let \( A(i) \) and \( B(i) \) be the AP and beam assignment for client \( i \). Ideally, to maximize network capacity via spatial reuse, Pia should maximize the signal-to-interference-ratio (SIR) of all clients:

\[
\max_{A,B} \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{BSM(A(i), i, B(i))}{\sum_{j=1}^{N_A} \max_{\tilde{B}}(B(j), i)}
\]

where \( BSM[\tilde{a}, i, \tilde{b}] \) denotes the signal strength from AP \( \tilde{a} \) to client \( i \) using beam \( \tilde{b} \), and \( \max_{\tilde{A}}(\tilde{a}, i) \) calculates the maximum interference that AP \( \tilde{a} \) can cause to client \( i \), which is given by:

\[
\max_{\tilde{A}}(\tilde{a}, i) = \max_{\tilde{A}(j)=\tilde{a}} \inf(j, i),
\]

\[
\inf(j, i) = \begin{cases} BSM[A(j), i, B(j)], & A(j) \neq A(i) \\ 0, & A(j) = A(i). \end{cases}
\]

However, directly solving Eq. (6) entails a high computational complexity of \( O((N_cN_A)^N_b) \), where \( N_b \) is the number of beam patterns in the phased array. This above SIR-based solution is not scalable to the client number. Moreover, in our evaluation (Sec. 5), we found it cannot process the pose input in real time for even two clients. Instead, we design a lightweight algorithm that maximizes the signal-to-leakage-ratio (SLR). The leakage is defined as the signal strength sent by an AP and received by an undesired client. SLR computes the ratio of desired client’s RSS to leakage. Maximizing the SLR will also enhance the SIR because it minimizes the leakage that causes interference to others. SLR reduces the optimization complexity by decomposing the dependency between clients in AP assignment and beam assignment.

(i) AP assignment. To maximize the SLR, it is best to assign the client to an in-FoV AP at closest distance, because it maximizes the RSS while minimizing the leakage from other APs. Pia thus assigns the client to an AP with the maximum average BSV across all beam indexes (e.g., client 1 is assigned to AP 1 in Fig. 12). When assigned to the same AP, multiple clients can work under the 802.11ad TDMA mode.

(ii) Beam assignment. Upon a new client assignment, the AP first iterates through each beam index, and calculates the RSS of current client and maximum leakage signal strength to other clients, which forms the SLR. The AP then chooses the beam assignment that has the maximum SLR value. Finally, it examines whether the resulting SIR for each client exceeds its packet decoding threshold; if not, it assigns a dedicated time slot for it. Algorithm 2 summarizes the SLR-based algorithm. Its runtime complexity can be straightforwardly derived as \( O(N_cN_b(N_s + N_c)) \), and it is efficient enough to run in real time. Although the SLR-based algorithm is suboptimal, our evaluation shows its performance is close to the SIR-based algorithm.

4.2 Dealing With the Environmental Reflectors

The above scheme geometrically models the interference coming from transmitters within direct LOS. In practice, when the transmitters become close to strong ambient reflectors, their beams may be redirected, causing NLOS interference. To account for such effects, Pia incorporates a novel reflector sensing scheme which enhances the model in Sec. 4.1.2. It requires the user to conduct a one-time environment learning, by placing the 60 GHz client device near each strong reflector (typically concrete walls and large metal furniture [34]). Due to the well-known sparsity of 60 GHz channel [35], from 60 GHz radio’s eyes, there are only a few dominant reflectors, which can account for more than 95% of the total energy that can be received by the radio in a typical indoor environment [36]. The selective scanning reduces the human labor and accelerates the sensing process by avoiding many unnecessary positions. However, we cannot let the user directly input the reflector coordinate since the reflectivity is unknown. Pia uses a novel signal angle sensing algorithm (Sec. 4.2.1), which fuses the pose sensor measurements with the BSV measurement, to estimate the relative location/orientation of reflectors. During the SLR estimation, Pia accounts for the NLOS interference from each reflector by modeling it as a virtual transmitter (Sec. 4.2.2). Moreover, strong reflectors unlikely move frequently due to the size and weight. We only need to rerun environment learning when their positions change.

4.2.1 Pose-Assisted Compressive Angle Estimation

The key idea in our NLOS interference model is to reverse engineer the impact of reflectors by tracing back the NLOS signals’ angle of departure (AoD) and angle of arrival (AoA). Such signal angles may be measured using a cone-shaped antenna with perfect
directionality [37]. However, our goal is to enable angle sensing in practical 60 GHz devices which have imperfect beam patterns\(^1\) (Fig. 11). The results are consistent with previous measurement study [25] using similar hardware. Our solution originates from a key observation: Whenever the reflector’s impact becomes non-negligible, the measured BSV will deviate from the one predicted by using the LOS model (Sec. 4.1.2), and become the superposition of two BSVs along the LOS path and reflection path (Fig. 13).

Following the same approach as Sec. 4.1.1, Pia first obtains the measured BSV, from which it then estimates the number of paths and their angles, based on a statistical optimization model. For simplicity of exposition, we focus on the AoD estimation, but the AoA applies in the same way. We first uniformly divide the AP-to-client view angles \((\theta, \phi)\) into \(N_p\) pairs following the geodesic grid [38]. Let \(\mathbf{V}(\theta, \phi)\) denote the predicted BSV (Sec. 4.1.1) for the \(i\)th pair of direction \((\theta_i, \phi_i)\), and \(g_i\) be the unknown channel gain along the BSV direction. The measured BSV \(\mathbf{V}_r\) can be expressed as the sum of BSVs from all directions multiplied by their associated gains: 

\[
\mathbf{V}_r = \sum_{i=1}^{N_p} \mathbf{V}(\theta_i, \phi_i) g_i.
\]

Thus, we may estimate the AoD by solving \(g_i\) in the system of equations:

\[
\mathbf{V}_r - \sum_{i=1}^{N_p} \mathbf{V}(\theta_i, \phi_i) g_i = 0 \tag{7}
\]

where the direction \((\theta_i, \phi_i)\) associated with a non-zero \(g_i\) will be the AoD of one path.

The problem \((7)\) is under constrained because \(N_p > N_b\). For example, to achieve an AoD estimation granularity of 8\(^\circ\), we need to uniformly divide the 3D view angle into 337 directions according to the geodesic grid [38]. Yet, for a phased-array of 32 elements, the number of beam patterns is only 64 following the standard practice of codebook design [39]. To address this problem, we harness the channel sparsity of the 60 GHz channel [35], i.e., most AoD directions have close to zero gain (i.e., \(g_i \approx 0\)), and only those corresponding to the LOS and a few strong reflection paths are non-negligible. We thus stack the \(g_i\) into an \(N_b \times 1\) column vector \(\mathbf{G} = [g_1, \ldots, g_{N_b}]^T\), which is now sparse. Thus, we can reformulate Eq. \((7)\) into a compressive sensing problem that minimizes the \(l_1\) norm of the sparse vector \(\mathbf{G}\):

\[
\min \|\mathbf{G}\|_1 \text{ subject to } \|\mathbf{V}_r - \hat{\mathbf{V}}\mathbf{G}\|_2 \leq \varepsilon \tag{8}
\]

where \(\hat{\mathbf{V}} = [\mathbf{V}(\theta_1, \phi_1), \ldots, \mathbf{V}(\theta_{N_p}, \phi_{N_p})]\) is an \(N_b \times N_p\) matrix, and \(\varepsilon\) is determined by the radio’s noise power in BSV measurement. Eq. \((8)\) can be solved efficiently in polynomial time using the \(l_1\)-MAGIC [40] or CVX [41] toolbox.

### 4.2.2 Modeling the Reflector Impact

\(^1\) Practical phased-array beam patterns deviate from the horn shape because: (i) The OEM codebook does not account for the electromagnetic impact from peripheral electronic components on the phased array. (ii) To reduce the manufacturing cost, the 60 GHz phased-arrays only have limited phase-shift resolution (2-bit for each antenna element on the Qualcomm platform we use, i.e., switching among 0\(^\circ\), 90\(^\circ\), 180\(^\circ\), and 270\(^\circ\)). (iii) Our radio uses the same beam pattern for both beacon frames and data frames. The former is expected to have more diverse and wider beams [5].

Given the AoA/AoD, Pia can back-trace the departure and arrival paths, and locate the intersection points as the estimated reflector positions. Let \(\mathbf{P}_e = [x_c, y_c, z_c, \theta_e, \phi_e]\) and \(\mathbf{P}_s = [x_s, y_s, z_s, \theta_s, \phi_s]\) be the poses of client and AP, and \((\theta_s, \phi_s)\) and \((\theta_e, \phi_e)\) be the estimated AoA and AoD. The AP’s pose can be estimated by the APS algorithm (Sec. 3.2). The AoA path travels along a series of points with coordinate:

\[
\mathbf{K}_a(l_a) = \mathbf{R}_r(\theta_e) \mathbf{R}_d(\phi_e) \begin{bmatrix} \cos(\phi_s) \sin(\theta_s) l_a \\ \sin(\phi_s) \sin(\theta_s) l_a \\ \cos(\theta_s) l_a \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{9}
\]

Eq. \((9)\) first rotates a point at the AoA angle \((\theta_e, \phi_e)\) w.r.t. the origin by an offset of client’s orientation \((\theta_s, \phi_s)\), and then translates the coordinate by an offset of client’s location \([x_c, y_c, z_c]\). The only unknown variable in Eq. \((9)\) is \(l_a\) that controls the distance between point \(\mathbf{K}_a(l_a)\) and \([x_c, y_c, z_c]\). In a similar way, we back trace the AoD path and form a series of points with coordinate \(\mathbf{K}_d(l_d)\) with \(l_d\) as the variable.

In practice, the AoA and AoD paths may not intersect in 3D space due to the residual error of angle sensing. Pia will select a point with closest distance to the two paths. Toward this goal, Pia determines the two points \(\mathbf{K}_a(l_a^\ast)\) and \(\mathbf{K}_d(l_d^\ast)\) with \(l_a^\ast\) and \(l_d^\ast\) that minimize the distance between them:

\[
(l_a^\ast, l_d^\ast) = \arg \min_{(l_a, l_d)} \|\mathbf{K}_a(l_a) - \mathbf{K}_d(l_d)\|_2. \tag{10}
\]

The closest intersection point, i.e., the estimated reflector position, \(\mathbf{K}_r\) is their median center: \(\mathbf{K}_r = (\mathbf{K}_a(l_a^\ast) + \mathbf{K}_d(l_d^\ast))/2\).

To capture the reflector’s impact, Pia adopts the concept of virtual AP (vAP) – a virtual signal source that mirrors the position of real AP relative to the reflector (Fig. 13). This idea is inspired by the 60 GHz ray-tracing method [42–44], which has proven to predict the 60 GHz signal propagation reasonably close to real measurement. A reflector will redirect the signals so that they look like emitting from the vAP. Given the client position \(\mathbf{K}_c = [x_c, y_c, z_c]^T\) and reflector position \(\mathbf{K}_r\), the vAP position \(\mathbf{K}_v\) can be derived from geometry (Fig. 13), by moving from the reflector position along the AoA direction over distance \(l_d^\ast\):

\[
\mathbf{K}_v = \frac{l_a^\ast}{l_d^\ast} (\mathbf{K}_r - \mathbf{K}_c) + \mathbf{K}_r. \tag{11}
\]

To determine the vAP’s orientation \((\theta_v, \phi_v)\), Pia solves following equation:

\[
\mathbf{R}_v(\theta_v) \mathbf{R}_d(\phi_v) \begin{bmatrix} \cos(-\phi_v) \sin(\theta_v) \\sin(-\phi_v) \sin(\theta_v) \\cos(\theta_v) \end{bmatrix} = \mathbf{K}_c - \mathbf{K}_r \frac{l_d^\ast}{l_a^\ast}, \tag{12}
\]

where the left side forms a unit vector from vAP’s orientation \((\theta_v, \phi_v)\) toward vAP’s AoD direction \((\theta_e, \phi_e)\). The unit vector should be at the same direction as the one from reflector’s position to client’s position (i.e., right side). The azimuth angle of vAP’s AoD is negative because it is the mirror symmetry of the real AP.

Once the pose of vAP is determined, it will be used in Pia’s spatial sharing module (Sec. 4.1.2) in the same way as a real AP, except that vAP should account for the reflection loss of reflector (estimated in the channel gain \(g_i\)). This procedure of vAP localization should be repeated for each reflector during the sensing stage.

---

\(\sum\)
4.3 Calibrating the Client’s Pose Offset

Since the outputs of APS and reflector sensing will be used by other client devices, in order for Pia to operate correctly, each later joined client that does not run the sensing stage, needs a one-time calibration for its pose. The pose information on mobile devices may contain an unknown initial offset w.r.t. the AP’s coordinate due to two reasons. (i) Not all devices are capable of reporting absolute location/orientation. Many accurate and mature indoor navigation technologies [18, 45–48] can only measure relative location/orientation change based on visual-inertial sensors. (ii) The pose, especially orientation, of a phased array antenna may not be the same as the pose of its host device.

Therefore, Pia needs to estimate an initial pose for each newly joined client w.r.t. the AP’s coordinate, so that it does not need to rerun the APS and reflector sensing. Our solution is designed upon the compressive angle estimation (Sec. 4.2.1), following two steps:

(i) Determining the in-FoV APs. When a client first enters the network area and moves around, it searches for two nearby APs within the FoV. There are multiple ways to determine if an AP is in the FoV. For example, the client can estimate the link throughput by sending data to the backend server and using the throughput thresholding heuristic in Sec. 2.1. Alternatively, the client can estimate the RSS from the beacon frames. Then, the client estimates the AoAs and AoDs w.r.t. the LOS APs (Fig. 14). Note that the sensed angle is relative to the device’s phased array antenna rather than the AP’s coordinate. Then, the two APs share their pose information (estimated from APS) with the client. (ii) Determining the initial pose. The two LOS paths originating from two APs intersect at the client’s location (Fig. 14). Given the APs’ locations and the AoD of the LOS paths, a simple geometrical calculation can pinpoint the client’s location. Pia can determine the client’s orientation in the AP’s coordinate, in a similar way as finding the vAP’s orientation in Sec. 4.2.2. The estimated location and orientation together form the initial pose of the client. It is worth noting that the initial pose estimation requires the AP fall in the client’s FoV. Thus, it cannot be applied to assist the network operations, such as AP switching.

5. IMPLEMENTATION

Multi-AP 60 GHz network testbed. We prototype Pia based on a multi-AP 60 GHz testbed, as shown in Fig. 15. The testbed consists of three APs and one client, all equipped with the Qualcomm 60 GHz wireless network interface card (with QCA6310 32-element phased array/RF front-end and QCA6320 MAC/baseband). Each AP is a small PC which interfaces the card through an M.2-to-PCIe adapter. We mount the phased array antenna on a flat panel that can adjust its tilt angle. Each client is a laptop (Acer P446-M59BB) with the phased array built in the outer side of the laptop’s cover.

Although the server-to-AP backhaul links can use 60 GHz interfaces with fixed beams, our prototype implements them using 10 Gbps Ethernet with a NETGEAR X5708E switch. We configure the two-hop network into two subnets, and the APs simply act as relays between client and the backend server (Fig. 15). To switch the AP, Pia modifies their gateway, and the data will be forwarded along the new AP. The switching latency is less than 5 ms (Sec. 6.2).

Extracting radio information from Linux driver. We use the latest wif6210 driver [49] for the 60 GHz wireless adapters. Since the firmware (wil6210.fw) and phased array codebook (wil7210.brd) are still unavailable in the Linux firmware library, we port them from a Windows driver (Atheros Sparrow 1.1ad version 3.3.3.7233) that is released by Acer. We modify the wif6210 driver and make it quickly (131 entries/second) export the low-layer information to user space, including instantaneous (Tx/Rx) throughput, signal quality, active (Tx/Rx) beam index, and (Tx/Rx) MCS. Such information is used as input to Pia’s modules.

Client pose. We attach a Google Tango tablet [18] to the laptop’s outer surface, which can track the laptop’s relative pose at cm-level and degree-level accuracy [50, 51] with less than 80 ms latency [52]. We will evaluate the impact of pose errors in Sec. 6. We develop an Android application that converts the quaternion rotation into pose (Sec. 3.1.1) and streams it to the laptop via USB at 297 entries/second.

Pia components. We implement the major design components (AP selection, APS, spatial sharing, signal angle sensing) in Matlab and C. The algorithms run on the back-end server (Fig. 1) which is a commodity PC with i7-3770k CPU and 16 GB memory. To achieve millisecond AP switching latency, many legacy 802.11 drivers (e.g., ath5k, ath9k, ath10k [53] and iwlib) support mac80211 VIF [54] which allows one client to maintain multiple virtual interfaces to different APs [55, 56], thus eliminating the re-authentication and re-association that account for the major overhead in AP switching. Since such support is still under development in wif6210, we achieve rapid AP switching through reverse-tethering, i.e., configuring the AP into the managed mode and client into the AP mode, which allows the laptop to maintain connections to multiple senders.

Measuring beam patterns. Pia’s interference management requires the phased array’s beam patterns (Sec. 4). Since our 60 GHz radio’s hardware specifications are not public, we measure them using the same 60 GHz software radio as in [35, 36, 57]. We place a receiver radio (with a 20° horn antenna) 2 m away from an AP and point its antenna center to the AP’s phased array. We mount the AP on a 3D motion controller that can rotate its azimuth/elevation angle at a step of 3°, allowing us to measure the RSS from different directions (i.e., angular beam pattern). Since the beam switching on the 60 GHz wireless card is controlled by closed-source firmware, we are currently unable to switch the phased array’s beam directly. In order to obtain the beam patterns for all beam indexes (31 in total), we measure the RSS values of beacon frames at the beginning of each beacon interval, which are sent sequentially by the AP, each using a different beam pattern, following the 802.11ad standard [5, 27]. We predict the BSV using pose data from the measured beam patterns. Besides, we find that in our platform the beacon frames share the same codebook as the data frames. Even though they may be different in other radios, we can still follow the same principle and measure beam patterns of the data frame.

6. EVALUATION

Methodology: We deploy the 60 GHz testbed in a 7 × 8 m² office environment, with 3 APs mounted on stands (2 m height) near the corner of the room, forming a triangle and facing 45° towards the ground. We choose the corner positions because of the physical constraints and convenience, just like the placement of typical

Figure 15: (Left) Pia’s hardware components, phased-array in the red circle. (Right) Multi-AP configuration.
home routers. Besides, it is also the best way to stress-test our system’s capability since the client needs to perform AP switching more frequently. Pia can easily scale to a larger space with more APs.

To bootstrap the system, we first generate a set of legitimate AP poses to feed APS (Sec. 3.2). Pia creates a bounding box using the coarse room size input by the user, and then partitions the possible AP coordinate by a step of 10 cm. Further increasing the granularity will not improve the AP-pose sensing accuracy because it is already close to the system limit (Sec 6.1). We then execute the AP switching in real time (Sec. 3.1). The current Qualcomm firmware does not support real-time beam control and RSS feedback for each beacon frame. To evaluate the interference management (Sec. 4.1), we collect BSV data over 100 random client poses for the commodity 60 GHz AP using our customized 60 GHz Rx (Sec. 5). We then use the BSV data as input to evaluate the performance of Pia’s spatial reuse, angle sensing, and reflector estimation, which best approximate the end-effect of an actual device.

**Metrics:** In the micro benchmarks, we focus on metrics directly related to the design modules’ performance, e.g., AP selection accuracy, pose error, SIR, and angle sensing error. The system level tests will focus on network-level metrics: throughput, latency, and reliability. We measure the achievable throughput by iperf3 [58] and the network latency by sending back-to-back ping packets. By default, error bars in all results denote the 90-percentage error.

### 6.1 Micro Benchmarks

#### 6.1.1 Effectiveness of APS and AP Selection

**Impact of training.** We first evaluate how the sensing stage affects APS (Sec. 3.2) by collecting training data while walking randomly over 15 meters, which only takes a user less than one minute to walk through. We partition the collected data into multiple sizes to evaluate the impact of data size on the sensing accuracy. Fig. 16 shows that both the location error and orientation error decrease dramatically over longer walking traces. When the walking distance is <5 meters, the estimation error tends to be large, because Pia doesn’t gather sufficient data to statistically filter out erroneous locations. With >11 m training, the AP’s pose error stabilizes at 0.29 m and 0.17 radian (i.e., 9.7°), which indicates the APS algorithm can quickly estimate AP’s pose with minimal training efforts.

We then use the collected data onwards for Pia’s link predictor. Fig. 17 shows that with distance >11 m, Pia’s AP selection reliability (i.e., choosing an in-FoV AP) can maintain at 94.4%. Pia can achieve a high prediction accuracy using relative small training efforts because it exploits the geometrical structure of the FoV.

**Impact of client’s pose information error.** We then inject Gaussian noise to the collected data, and vary the mean from mm to decimeter-level, consistent with the range of pose errors on mobile devices (Sec. 7). The resulting APS location estimation error (Fig. 18) grows linearly with the client’s pose error. Fortunately, since many recent mobile location tracking schemes [59–61] can already provide at least 10~30 cm accuracy, the estimated AP pose is sufficient for reliable AP selection. Besides, since the APS is a one-time initialization, specialized tracking devices such as Tango can be used to ensure high precision. In addition, the APS orientation is not sensitive to client’s pose error because the sweet spot of ellipsoid center (Sec. 3.2) averages out the deviation.

Fig. 19 plots the impact of pose error on link prediction. The location error has a limited impact—even for 1.21 m client position error, Pia’s link prediction accuracy only decreases by 2.1%. Yet, 0.52 radians (30°) of orientation error reduces the prediction accuracy by more than 10%, because orientation error more easily deviates the radio’s FoV. Fortunately, recent 3D orientation tracking systems can maintain the error below <5° [62], as accurate as Tango. Therefore, Pia can run on mobile devices with more than 90% AP selection accuracy.

**AP switching latency.** To verify if the client can seamlessly switch across APs, we force it to switch between two APs per second. Fig. 21 plots the round-trip latency between the client and backend server, from which we observe three patterns. The latency is less than 1 ms when measure occurs in a beacon interval (BI), and increases to 3 ms at the beginning of BI due to the beaconing overhead. The 3 ms latency follows the 100 ms periodicity that well matches with the AP’s beacon interval. The AP switching causes additional 10~25 ms packet latency, because of the beam training overhead with the new AP. Yet, every time it only affects a single packet. Besides, the pose information can be exploited to reduce the beam adaptation overhead, which we leave for future work.

#### 6.1.2 Effectiveness of Compressive Angle Estimation and Interference Prediction

**Accuracy of signal angle sensing.** The performance of Pia’s compressive angle estimation (Sec. 4.2) is determined by the SNR of BSV measurement. To evaluate the impact, we first measure BSV under higher SNR (>30 dB). Since BSVs are obtained by measuring the detailed waveform of beacon frames, we can add Gaussian noise to the original waveform and create BSVs for different SNRs. The CDF plot in Fig. ?? shows an average angle estimation error of less than 5° when SNR is high (e.g., >9 dB). Residual errors are caused by discretization of the view angles (Sec. 4.2). The angle error may drift a lot under low SNR, when the BSV is significantly distorted by noise. However, by simply repeating the BSV measurement by 5 times and averaging out, angle error can be reduced to less than 5° (Fig ??). Such repetitive measurements add little extra sensing time (only 5 BIs for 5 measurements).

Pia’s angle sensing can also estimate each path’s channel gain (Sec. 4.2.1). Fig. 22 plots the CDF of estimation error, where we obtain the ground-truth by dividing the measured BSV by the predicted one (Sec. 4.1.1). A higher SNR leads to more accurate gain estimation (error <3.4 dB when SNR >9 dB). Notably, the estimated gain is generally smaller than the ground-truth because compressive sensing solver spreads the energy to other directions to minimize the residual error.

**Accuracy of client-pose and reflector estimation.** Since the client’s pose estimation and reflector estimation rely on the compressive angle estimation, we run them consecutively. We first repeat the former at 20 random spots and compare it with ground-truth that measures w.r.t. AP’s coordinate using a laser range-finder. The result (Fig. 23) shows a mean position offset of 0.5 m. We further place a strong reflector (metal sheet) and randomly vary its position around the Tx/Rx from 1 to 3 m. The reflector position estimation shows similar accuracy, albeit requiring 5 BSV measurements due to low SNR (reflected signal is 10~15 dB weaker than LOS). Notably, the positioning accuracy is comparable to state-of-the-art indoor localization systems [45, 60, 61, 63], and the orienta-
6.2 System Level Tests

6.2.1 Network Robustness by AP Switching

We conduct a system-level test of the AP switching by emulating the wireless VR scenario, where the client follows a random walk and pan/tilt change. The backend server streams a real-time uncompressed video (1280×720 at 75 FPS, approximately 1.58 Gbps bitrate) via the APs to clients. Owing to the channel fading, throughput measurement will show small fluctuations even when the link is stationary. The small throughput variations do not affect the video transmission reliability, but will falsely trigger many bad link indications if we threshold the link state by the video bitrate. Thus, we choose the threshold as 95% of the video bitrate and consider video frames cannot be delivered reliably when the network throughput drops below $T = 1.50$ Gbps. We run Pia against 4 schemes: (i) noSwitch: Client connects to a single fixed AP. (ii) hardProbe: Client sequentially probes each AP (once per second) by temporarily switching to it and sending a few packets to have the MCS stabilized. If the probed AP’s link quality is better, the client will stay with it. (iii) softProbe: Client measures network throughput every 0.5s, and probes for the best one only if the throughput drops below $T$. (iv) oracle: Switch AP based on the known throughput in the trace data without counting any overhead.

Link availability. Fig. ?? plots the link availability, i.e., percentage of time that throughput exceeds $T$. Pia’s availability is 97.3%, close to the oracle (99.6%) whose availability is below 100% due to certain blind spots not covered by any AP. This can be addressed by proper deployment or adding more APs. The availabilities of other schemes are much lower (<84.9%). Even hardProbe and softProbe show low availability due to the AP probing overhead.

Hazard times. Fig. ?? plots hazard times, i.e., the number of occurrences that link throughput drops below $T$ in a 5-minute test. Compared with noSwitch, softProbe, and hardProbe, Pia reduces the hazard times by 6.3×, 4.5× and 4.2×. The extra hazard times over the oracle is caused by occasional wrong AP selection. The unavailability, i.e., 1-availability, divided by hazard times gives the average duration of each drop. Pia can not only quickly recover from wrong predictions, but also prevent most throughput drops by proactively switching the AP.

Video frame latency. Fig. 27 plots the measured video frame latency. The mean and 90-percentile latency of Pia (11.0 ms and 17.8 ms) are close to the oracle (10.6 ms and 16.4 ms), and much lower other schemes (90-percentile about 40 ms). Note that the latency can be reduced substantially under higher PHY bit-rates (802.11ad supports up to 6.7 Gbps, whereas the Qualcomm radio can only reach 2.5 Gbps).

6.2.2 Multi-client Spatial Sharing

We conduct trace-based emulation to evaluate Pia’s spatial sharing mechanism. We reuse the BSV trace collected in Sec. 6.1, calculate the SIRs of clients in concurrent transmission, and map them to the achievable bitrate following a standard 802.11ad rate table [35]. The impact of reflectors is modeled through the vAPs (Sec. 4.2). We determine the AP and beam selection for each client using Pia and 3 baselines: (i) noShare: The transmission opportunity is randomly assigned to one client at a time. Each client finds the best AP and beam based on the 802.11ad AP discovery protocol, which may not always be the best choice. (ii) oracle: The optimal AP and beam are determined for each client. This is unrealistic, but serves as a benchmark. (iii) softProbe: Client measures network throughput every 0.5s, and probes for the best one only if the throughput drops below $T$. The protocol does not take into account the reflection effects between the client and the AP. The proposed Pia outperforms the baselines in terms of the achievable bitrate and the number of retransmissions, while maintaining low latency and high availability.
and beam training. (ii) 802.11ad: Clients run interference sensing, each in a dedicated BI, to determine the concurrent transmission feasibility (Sec. 4.1). Afterward, they transmit concurrently in next 5 BIs. (iii) oracle: Maximizing the clients’ SIRs (Eq. (6)).

Throughput gain. Fig. 28 (a) plots the achievable throughput over the number of clients. All clients achieve similar throughput in noShare, since it roughly splits the time equally. 802.11ad improves throughput by 17.3% compared to noShare. Yet, clients suffer from unfairness—client1’s throughput is 75.7% higher than others, because client1 is at a position causing asymmetric interference that starves others, a well-known problem in directional networks. Pia performs closely to the oracle, with 1.47 × gain over 802.11ad for 2 and 3 client cases, respectively, and offers higher fairness. Note that the throughput stops increasing as client number reaches 4 because spatial reuse saturates.

Concurrent transmission opportunity. Fig. 28 (b) plots percentage of successful concurrent transmissions for client1 and client2. Although the percentage gradually decreases as more clients join, Pia improves the concurrent transmission opportunity by 28% and 36.7% for 2-client and 4-client respectively compared to 802.11ad. The stacked bars represent a breakdown of the source of gain, from selecting beams with weaker signal leakage (bottom) and eliminating interference sensing overhead (top), which together verify the effectiveness of Pia’s pose-assisted spatial reuse mechanism.

7. DISCUSSION
60 GHz AP deployment. Based on our experiments, we identified 3 general guidelines to maximize the coverage and effectiveness of Pia in a multi-AP network. (i) Deploying the APs higher than typical human heights, such as ceiling, that will maximize the AP’s FoV to the client and minimize blockage of other passing-by people. (ii) However, placing the AP to the ceiling cannot prevent the link blockage. When a user walks away from the AP’s beneath center, the client’s FoV is still prone to moving out of the AP’s FoV. To ensure multi-Gbps coverage everywhere, the APs should be densely deployed. (iii) The FoV of APs should be partially overlapped to minimize the number of blind spots. Besides, it is worth noting that Pia is not tied to any specific ways of AP deployment.

Pose information availability. Our evaluation results show Pia can tolerate 0.5 m location error and 0.35 radians orientation error. Current mobile devices, combining motion sensors, light sensor, camera, etc., can already provide accurate pose tracking. A2° [62] can estimate 3D orientation <5° using accelerometer and magnetometer. Many mobile localization systems [45–48, 66] can track users at centimeter to decimeter accuracy. Commercial products such as Google Tango [18] can provide the 6-DoF tracking at cm-level accuracy, by using motion sensors and a depth-camera. VR headsets like HTC Vive [17] can tracking user motion at mm-level precision [67], which is the ideal candidate for our system. All of these systems suffice to support Pia’s pose-assisted design principle.

Extension to general mobile 60 GHz networks. In this work, we tailored the system design to match a typical wireless VR setup in a constrained room environment. However, the design principles can be extended to general millimeter-wave networks to support mobility and seamless coverage. Although the absolute location tracking technologies for mobile devices may not be sufficiently accurate and reliable, the relative motion tracking (e.g., rotation change and moving offset) using built-in visual-initial sensors can suffice for Pia. To alleviate the accumulated drift of relative tracking, Pia may periodically apply the pose calibration (Sec. 4.3). Moreover, to mitigate the initial training overhead, we could deploy APs to known locations and develop an online-sensing method to infer the phased array’s orientation (inside the AP) and reflectors’ poses during usage. Though it may trigger many error predictions at the beginning, the accuracy could improve over more collected data, which we will leave for our future work.

8. RELATED WORK
Pia is most closely related with the following domains:

Robust 60 GHz networks. To make 60 GHz networks robust against human blockage and movement, prior research primarily focused on minimizing the beam searching overhead, so that the 60 GHz links can efficiently recover from disruption. Existing systems leveraged the correlation between beams [68] or used out-of-band channel [2, 69] to estimate the best beam at low overhead. However, when blockage occurs, the reflection path may be either weak or non-existent [35]. To ensure better coverage, MoVR [2] adopted a customized 60 GHz relay to amplify and forward the AP’s signals. To maximize the beam alignment, MoVR also leverages the pose to guide the AP/mirror’s beam steering. Pia shares similar spirit—the multiple APs act as relays for the client. Yet, Pia focuses on the more general problems of AP selection/switching and spatial reuse, harnessing the pose information on client devices. Pia is compatible with 802.11ad or any future 60 GHz base stations, and does not need dedicated analog signal forwards as in [2].

The general idea of using smartphone sensors to improve network performance has been examined in [70], which proposed to
synthesize various motion sensor hints to adapt protocol primitives such as bit-rate adaptation. Sani et al. [71] conducted an empirical study and showed that, by judiciously choosing among multiple directional antennas on a mobile device based on motion sensor information, the WiFi link SNR can be improved by 3 dB. In contrast, new challenges emerge in 60 GHz networks due to vulnerability to blockage and limited FoV. Meanwhile, the electronically steerable beams bring new opportunities to spatial reuse. Yang et al. [72] designed a sensor-assisted multi-level codebook for efficient beam searching under mobility. Pia, in contrast, harnesses the 5-DoF pose information to address the network-level problems arising from multiple APs and clients.

AP handover and selection. Numerous solutions have been proposed to curtail the AP handover/reassociation overhead in WiFi networks. The most viable idea is to fork multiple virtual network interfaces [55, 56]. Although the switching mechanism itself is still applicable, the decision metrics for switching (e.g., signal strength, distance [13, 14] and location [12, 15, 16]) in low-frequency networks are no longer applicable to the 60 GHz network due to orientation sensitivity. Alternative machine-learning based selection algorithms [73–75]) have been explored. However, such learning algorithms require dense training, and the accuracy is still much lower compared with Pia’s geometry-model based algorithm [6]. Athanasiou et al. [76] proposed an asymptotically optimal algorithm to balance and ensure fair client association in 60 GHz networks. In contrast, Pia is the first work from an architecture/systems perspective that optimizes the 60 GHz link quality and reliability via AP/beam switching using the pose information.

Spatial reuse of 60 GHz networks. Directional antennas are known for achieving spatial reuse and alleviating interference. A vast literature [20] explored the MAC issues in directional networks. DIRC [65] further accounted for the impact of irregular interference with imperfect directionality. These systems focus on static radios, with a cone-shaped antenna model, which differs dramatically from the beam shape of practical 60 GHz phased arrays (Sec. 4.1). Park et al. [77] proposed a null-forming technique that can create specific null points to avoid interfering peer devices, but this requires arbitrary control of the antenna weights, whereas practical 60 GHz phased arrays can only use a set of fixed beamforming weights which form a codebook.

9. CONCLUSION

Whereas the “laser-like” beam of millimeter-wave radios has triggered many doubts about their coverage, we have demonstrated the high potential of Pia to address the concern. Pia employs the 5-DoF pose information on mobile devices, together with a multi-AP network architecture, to achieve seamless coverage under mobility and user’s orientation change. Pia can be applied to meet the needs of demanding Gbps applications such as wireless VR. Its design principle of pose-assisted mobility 60 GHz networking can also be extended to general mmWave pico-cell/small-cell networks with dense AP deployment.

ACKNOWLEDGEMENTS

We appreciate the insightful feedback from the anonymous reviewers and our shepherd Professor Haitham Hassanieh who helped improving this work. This project is supported in part by the NSF under Grant CNS-1506657, CNS-1518728, CNS-1343363, CNS-1350039, and CNS-1617321.

10. REFERENCES


[22] J. Zhang, X. Zhang, P. Kulkarni, and P. Ramanathan,


