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

ABSTRACT

60 GHz millimeter-wave networking has emerged as the next frontier technology to provide multi-Gbps wireless connectivity. However, the intrinsic directionality and limited field-of-view of 60 GHz antennas make the links extremely sensitive to user mobility and orientation change. Hence, seamless coverage, even at room level, becomes challenging. In this paper, we propose Pia, a robust 60 GHz network architecture that can provide seamless coverage and mobility support at multi-Gbps bitrate. Pia comprises multiple cooperating access points (APs). It leverages the pose information on mobile clients to proactively select the AP and manage multi-link spatial reuse. These decisions require a model of the pose/location of the APs and ambient reflectors. We address these challenges through a set of AP-pose sensing and compressive angle estimation algorithms that fuse the pose measurement with link quality measurement on the client. We have implemented Pia using commodity 60 GHz platforms. Our experiments show that Pia reduces the occurrence of link outage by 6.3× and improves the spatial sharing capacity by 76%, compared to conventional schemes that only use in-band information for adaptation.

Keywords

60 GHz; Multi-AP; Mobility; Location; Orientation; Pose sensing

1. INTRODUCTION

The millimeter-wave (mmWave) wireless technology is emerging as a disruptive networking paradigm to provide multi-Gbps connectivity for demanding applications, such as wireless backhaul [1], cordless virtual reality (VR) [2], wireless fiber-to-home [3], mobile-to-screen video cast [4], *etc.* FCC's recent policy to release 14 GHz of unlicensed spectrum, along with standardization activities such as the IEEE 802.11ad [5], 802.15.3c [6] and 802.11ay [7], have spawned many consumer grade mobile devices on the 60 GHz mmWave band. For example, TPCAST [8], a 60 GHz adapter, can replace the cables between a VR headset and its PC host. Recent 60 GHz capable smartphones and laptops [9] can stream Gbps uncompressed videos to an external display.

Ideally, one would anticipate the 60 GHz technology as the next-frontier for mobile broadband, to replace the current WiFi. However, to date, the use cases of the 60 GHz wireless technology have 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.

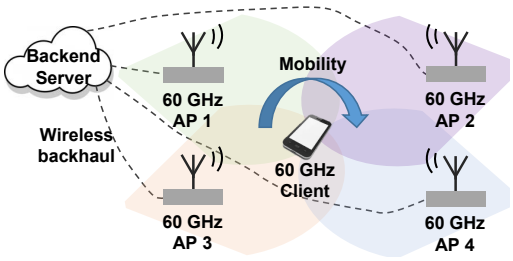


Figure 1: Multi-AP networks: APs are partially overlapped and connected to a backend server via wireless backhaul.

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.

MobiCom '17, October 16–20, 2017, Snowbird, UT, USA

DOI: https://doi.org/10.1145/3117811.3117832

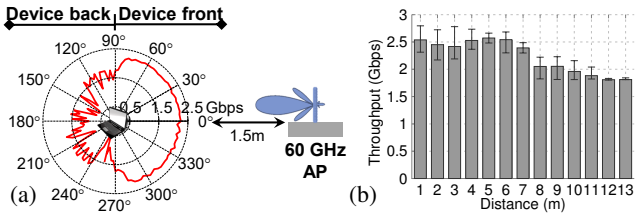


Figure 2: FoV and performance of the commercial 60 GHz laptop in our measurement. Varying (a) client device's azimuth orientation, (b) AP-to-client distance (azimuth = 0°).

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∼6.3×, and the spatial sharing enhances the network capacity by 76% compared to the 802.11ad.

2. BACKGROUND AND OVERVIEW

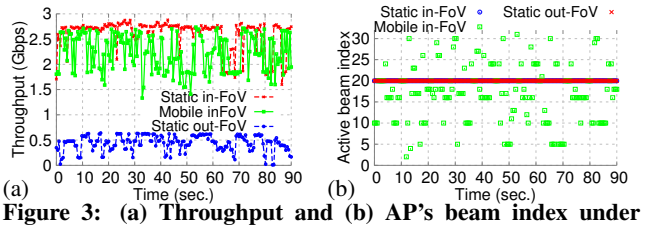


Figure 3: (a) Throughput and (b) AP's beam index under static/mobile in-FoV and static out-FoV.

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-

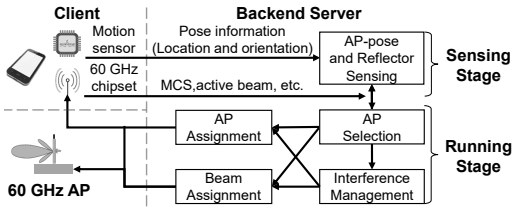


Figure 4: Pia's modules and operation workflow.

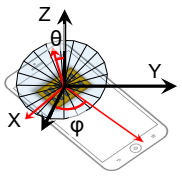


Figure 5: 5-DoF pose of the phased array.

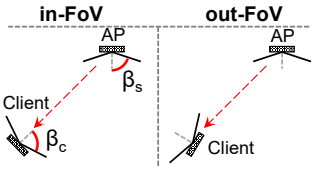


Figure 6: Link status prediction based on the poses.

tain 50% at worst and 80% on average compared to the stationary case, and 3∼5× 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 $\mathbf{P} = [x, y, z, \theta, \varphi]$, 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 the antenna norm to the edge of its coverage (β_s and β_c in Fig. 6). For example, the FoV of our device is 85°, measured in Fig. 2. A device's FoV is generally available from its antenna specifications, or can be manually measured following Sec. 2.1.

Let $\mathbf{P_c} = [x_c, y_c, z_c, \theta_c, \varphi_c]$ and $\mathbf{P_s} = [x_s, y_s, z_s, \theta_s, \varphi_s]$ be poses of client and AP, respectively. Pia computes two vectors $\mathbf{V_s}$ and $\mathbf{V_{sc}}$ that point to the norm of AP's antenna and the direction from AP to client, following 3D geometry:

$$\mathbf{V_s} = \mathbf{R_y}(\theta_s)\mathbf{R_z}(\varphi_s)[0, 0, 1]^{\mathrm{T}} \quad (1)$$

$$\mathbf{V_{sc}} = [x_c, y_c, z_c]^T - [x_s, y_s, z_s]^{\mathrm{T}} \quad (2)$$

where $(\cdot)^{\mathrm{T}}$ denotes transpose operator. $\mathbf{R_y}(\alpha)$ and $\mathbf{R_z}(\alpha)$ are the 3D rotation matrices [28] that rotate a point relative to the origin along Y-axis and Z-axis by an angle of α respectively. Eq. (1) converts the orientation of AP's antenna into a vector format. Then Pia can determine the angle ϕ_s between vectors $\mathbf{V_s}$ and $\mathbf{V_{sc}}$ by:

$$\phi_s = atan2(||\mathbf{V_s} \times \mathbf{V_{sc}}||_2, \mathbf{V_s}^{\mathrm{T}}\mathbf{V_{sc}}) \quad (3)$$

where function *atan2(·)* calculates the four-quadrant inverse tangent [29]. Similarly, Pia computes the angle ϕ_c between the norm of client's antenna and $-\mathbf{V_{sc}}$, and checks if ϕ_s and ϕ_c are smaller than the AP's FoV β_s and client's FoV β_c:

$$\text{Link\_State} = \begin{cases} 1, & \phi_s \le \beta_s \quad \phi_c \le \beta_c \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

Fig. 7 plots an example trace of the link predictor, as the client walks within a 2×3 m^2 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

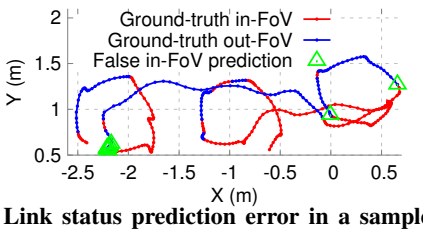


Figure 7: Link status prediction error in a sample walking: false in-FoV prediction is highlighted by green triangle. The dense cluster implies multiple errors occur consecutively.

the 1324 pose samples. More importantly, the errors mostly concentrate on the boundary area *i.e.*, when transiting 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 Acceleration (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 $\hat{\mathbf{P}}_\mathbf{c}$ at time $t+1$ can be predicted as:

$$\hat{\mathbf{P}}_\mathbf{c}(t+1) = \mathbf{P}_\mathbf{c}(t) + \Delta\hat{\mathbf{P}}_\mathbf{c}(t),$$
$$\Delta\hat{\mathbf{P}}_\mathbf{c}(t+1) = \Delta\mathbf{P}_\mathbf{c}(t),$$

where $\Delta\mathbf{P}_\mathbf{c}(t) = \mathbf{P}_\mathbf{c}(t+1) - \mathbf{P}_\mathbf{c}(t)$ 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).

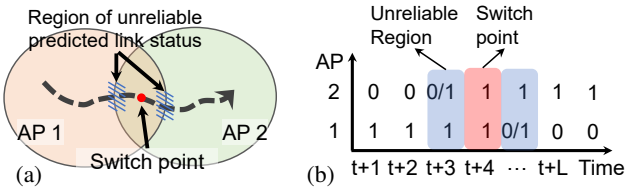


Figure 8: (a) AP switching during the transition area. (b) Table of predicted link status for two APs.

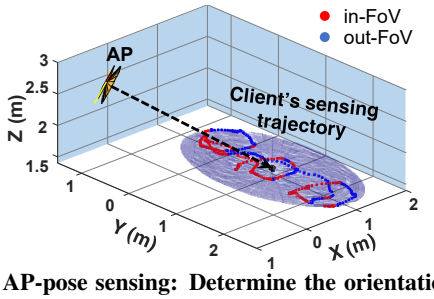


Figure 9: AP-pose sensing: Determine the orientation from a 3D ellipsoid that covers all client's poses in the AP's FoV.

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 $\mathbf{S_s}$ denote a set that contains all legitimate AP's pose values, and $\mathbf{S_t}$ 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 $\mathbf{P}^*$ 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

procedure *APposeSensing*($\mathbf{S_t}$, $\mathbf{T_t}$, w_{FI}, w_{FO})
  $[x_d, y_d, z_d]$ = ellipsoidCenter($\mathbf{S_t}$) ▷ Find the sweet spot
  $\mathbf{D}$ = APLocList($\mathbf{S_t}$) ▷ Generate list of legitimate AP locations
  vMin $\leftarrow \inf$, $\mathbf{P}^* \leftarrow$NULL ▷ Initialize variables
  for $[x_s, y_s, z_s] \in \mathbf{D}$ **do**
    $[\theta_s, \varphi_s]$ = toAngle($[x_s, y_s, z_s]$, $[x_d, y_d, z_d]$) ▷ To orientation
    $\mathbf{P}_i \leftarrow [x_s, y_s, z_s, \theta_s, \varphi_s]$, $v \leftarrow 0$ ▷ Candidate AP pose
    for $\mathbf{P}_j \in \mathbf{S_t}, t_j \in \mathbf{T_t}$ **do**
      $v = v +$ *inFoV*$(\mathbf{P}_i, \mathbf{P}_j) * (1 - t_j) * w_{\text{FI}}$
        $+(1 -$ *inFoV*$(\mathbf{P}_i, \mathbf{P}_j)) * t_j * w_{\text{FO}}$ ▷ Eq. (5)
    end for
    if $v <$ *vMin* **then** ▷ Minimize weighted error
      vMin $\leftarrow v$, $\mathbf{P}^* \leftarrow \mathbf{P}$
    end if
  end for
  return $\mathbf{P}^*$ ▷ Return the estimated AP's pose
end procedure

$$\mathbf{P}^* = \underset{\mathbf{P}_i \in \mathbf{S_s}}{\arg\min}\ w_{\text{FI}} \sum_{j=1}^{|\mathbf{S_t}|} \textit{inFoV}(\mathbf{P}_i, \mathbf{P}_j) * (1 - t_j) + w_{\text{FO}} \sum_{j=1}^{|\mathbf{S_t}|} (1 - \textit{inFoV}(\mathbf{P}_i, \mathbf{P}_j)) * t_j, \quad (5)$$

where $|\mathbf{S_t}|$ represents the number of samples in $\mathbf{S_t}$, and t_j is the ground-truth link status (0/1 value). Function *inFoV*$(\cdot)$ determines if the client and AP are within each other's FoV (Eq. 4) and returns a binary (0/1) indicator. $\mathbf{P}_i$ 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_{FI}:w_{FO} = 4:1 in Pia).

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

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 beamforming 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-

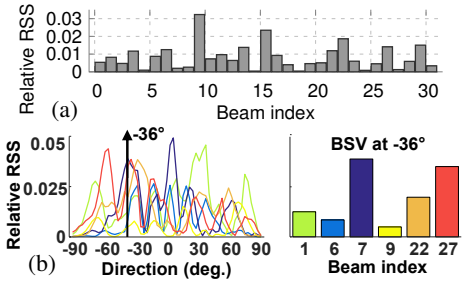


Figure 10: (a) Example measured BSV (RSS across all beam patterns for a given angle.) (b) Derive BSV from beam patterns at -36°.

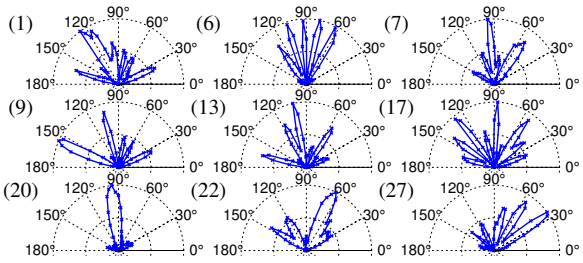


Figure 11: Samples of measured beam patterns of the phased array antenna. Left-top number is the beam index.

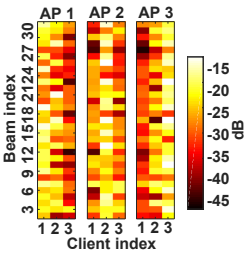


Figure 12: An example BSM for 3 APs and 3 clients.

sured 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_s APs. Let $\mathbf{A}(i)$ and $\mathbf{B}(i)$ 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:

$$\max_{\mathbf{A},\mathbf{B}} \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{BSM[\mathbf{A}(i), i, \mathbf{B}(i)]}{\sum_{j=1}^{N_s} INF_{max}(j,i)}, \tag{6}$$

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

$$INF_{max}(\hat{a}, i) = \max_{j, \mathbf{A}(j)=\hat{a}} INF(j, i),$$

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

However, directly solving Eq. (6) entails a high computational complexity of $\mathcal{O}((N_s N_b)^{N_c})$, 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 $\mathcal{O}(N_c N_b (N_s + N_c))$, 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.

Algorithm 2 SLR-based AP and Beam Assignment

procedure *assignSLR*($BSM[]$, N_c)
  for $i = 1 : N_c$ **do**
    $\mathbf{A}(i) \leftarrow indexOfMax(mean(BSM[:, i, :].'))$ ▷ Assign AP
  end for
  for $i = 1 : N_c$ **do**
    for $j = 1 : N_c$ **do**
      $SLR[j, :] \leftarrow \frac{BSM[\mathbf{A}(i), i, :]}{BSM[\mathbf{A}(i), j, :]}$ ▷ Leakage from client i to j
    end for
    $\mathbf{B}(i) \leftarrow indexOfMax(min(SLR[:, :]))$ ▷ Assign beam
  end for
  return $\mathbf{A}$ and $\mathbf{B}$ ▷ Return AP and beam assignment
end procedure

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

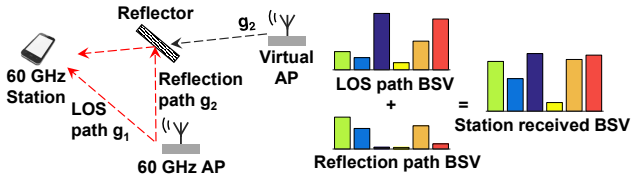


Figure 13: Compressive angle sensing and reflector estimation: measured BSV is the sum of BSVs along two paths.

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 (θ, φ) into N_p pairs following the geodesic grid [38]. Let $\mathbf{V}(\theta_i, \varphi_i)$ denote the predicted BSV (Sec. 4.1.1) for the i^{th} pair of direction (θ_i, φ_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: $\mathbf{V}_r = \sum_{i=1}^{N_p} \mathbf{V}(\theta_i, \varphi_i) g_i$. Thus, we may estimate the AoD by solving g_i in the system of equations:

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

where the direction (θ_i, φ_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°, 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 $N_p \times 1$ column vector $\mathbf{G} = [g_1, \ldots, g_{N_p}]^T$, 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 $\mathbf{G}$,

$$\min ||\mathbf{G}||_1 \quad \textit{subject to} \quad ||\mathbf{V}_r - \hat{\mathbf{V}}\mathbf{G}||_2 \leq \varepsilon \tag{8}$$

where $\hat{\mathbf{V}} = [\mathbf{V}(\theta_1, \varphi_1), \ldots, \mathbf{V}(\theta_{N_p}, \varphi_{N_p})]$ is an $N_b \times N_p$ matrix, and ε 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°, 90°, 180°, and 270°). *(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].

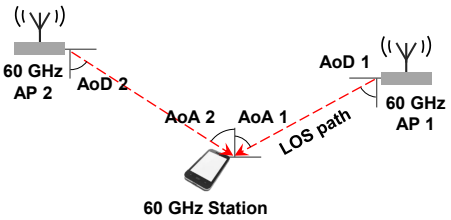


Figure 14: Estimate the initial pose of the client using compressive angle sensing.

Given the AoA/AoD, Pia can back-trace the departure and arrival paths, and locate the intersection points as the estimated reflector positions. Let $\mathbf{P_c} = [x_c, y_c, z_c, \theta_c, \varphi_c]$ and $\mathbf{P_s} = [x_s, y_s, z_s, \theta_s, \varphi_s]$ be the poses of client and AP, and (θ_a, φ_a) and (θ_d, φ_d) 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:

$$\mathbf{K_a}(l_a) = \mathbf{R_y}(\theta_c)\mathbf{R_z}(\varphi_c) \begin{bmatrix} \cos(\varphi_a)\sin(\theta_a) l_a \\ \sin(\varphi_a)\sin(\theta_a) l_a \\ \cos(\theta_a) l_a \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{9}$$

Eq. (9) first rotates a point at the AoA angle (θ_a, φ_a) *w.r.t.* the origin by an offset of client's orientation (θ_c, φ_c), 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 might 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 $\mathbf{K_a}(l_a^*)$ and $\mathbf{K_d}(l_d^*)$ with l_a^* and l_d^* that minimize the distance between them:

$$(l_a^*, l_d^*) = \arg\min_{(l_a, l_d)} ||\mathbf{K_a}(l_a) - \mathbf{K_d}(l_d)||_2. \tag{10}$$

The closest intersection point, *i.e.*, the estimated reflector position, $\mathbf{K_r}$ is their median center: $\mathbf{K_r} = (\mathbf{K_a}(l_a^*) + \mathbf{K_d}(l_d^*))/2$.

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^*:

$$\mathbf{K_v} = \frac{l_d^*(\mathbf{K_r} - \mathbf{K_c})}{l_a^*} + \mathbf{K_r}. \tag{11}$$

To determine the vAP's orientation (θ_v, φ_v), Pia solves following equation:

$$\mathbf{R_y}(\theta_v)\mathbf{R_z}(\varphi_v) \begin{bmatrix} \cos(-\varphi_d)\sin(\theta_d) \\ \sin(-\varphi_d)\sin(\theta_d) \\ \cos(\theta_d) \end{bmatrix} = \frac{\mathbf{K_c} - \mathbf{K_r}}{l_a^*}, \tag{12}$$

where the left side forms a unit vector from vAP's orientation (θ_v, φ_v) toward vAP's AoD direction $(\theta_d, -\varphi_d)$. 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.

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 XS708E 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 wil6210 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 11ad version 3.3.3.7233)

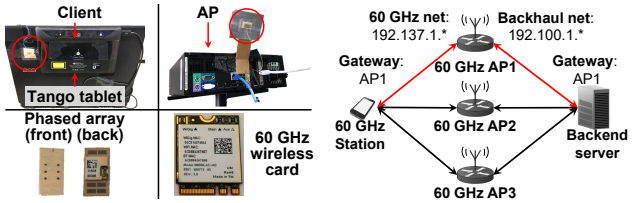


Figure 15: (Left) Pia's hardware components, phased-array in the red circle. (Right) Multi-AP configuration.

that is released by Acer. We modify the wil6210 driver and make it quickly (131 entries/second) export the low-layer information to user space, including instant (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 iwlwifi) 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 wil6210, 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 m^2 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

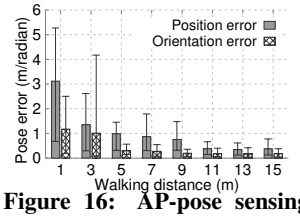


Figure 16: AP-pose sensing over user's training effort—walking distance.

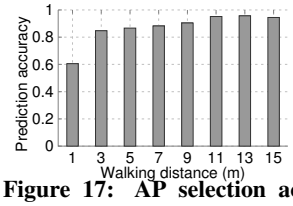


Figure 17: AP selection accuracy over user's training effort—walking distance.

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 rouge 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-percentile 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$\sim$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$\sim$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$\sim$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-

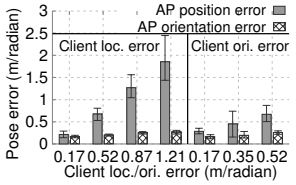


Figure 18: AP-pose sensing over client's pose error.

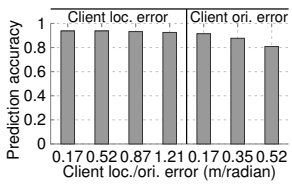
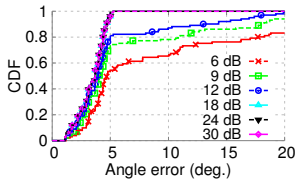


Figure 19: AP selection accuracy over client's pose error.



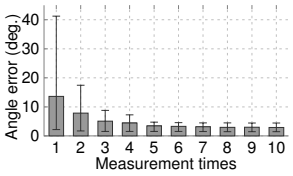


Figure 20: Angle sensing error (a) over the SNR of BSV, (b) using multiple BSVs (SNR=6 dB).

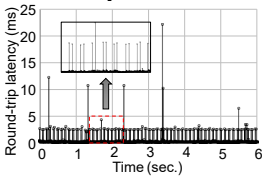


Figure 21: Packet latency during the AP switching.

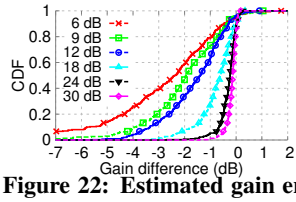


Figure 22: Estimated gain error over SNR of BSV.

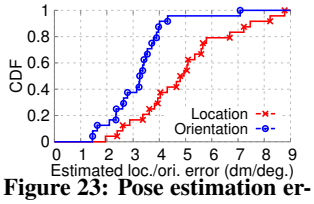


Figure 23: Pose estimation error using angle sensing.

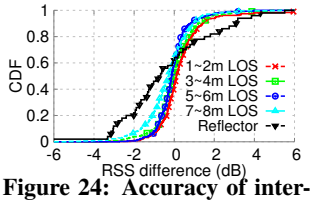


Figure 24: Accuracy of interference prediction.

tion accuracy $< 5^\circ$ suffices for Pia to run reliably (Fig. 19).

Interference prediction accuracy. Pia estimates the interference using predicted BSV (Sec. 4.1.2). To verify its accuracy, we calculate the signal strength difference between the collected BSV data and the predicted one subtracted by the pathloss of Frii's model (Sec. 4.1.2). We found Pia can reliably predict the strength of LOS paths based on their poses and active beam indexes regardless of the client-to-AP distance, with a small average error of 0.74 dB (Fig. 24). The prediction error over reflected paths is larger (<4 dB) due to extra errors from reflector estimation. Yet, Pia can tolerate this residual prediction error by using a more conservative decodable threshold (Sec.4.1.2).

Performance of SLR-based assignment. The SLR-based algorithm evades exponential searching space for optimal AP and beam assignment (Sec. 4.1.2). Fig. **??** and Fig. **??** plot the SIR and Jain's fairness [64] difference between the SLR algorithm and the optimum (*i.e.*, maximizing average SIR). We found at least 50% of the SLR outputs are the optimal assignment (*i.e.*, zero SIR difference), and 90% have <3 dB difference, implying the SLR-based algorithm performs close to the optimum. Furthermore, for those suboptimal assignments, the SLR-based algorithm more likely gives better fairness (*i.e.*, negative fairness difference), because the optimal solution may sacrifice fairness to maximize the SIR.

Since we do not have real-time control over the radio's transmit beam index, we cannot quantify the impact of client's moving speed. The beam selection (Sec. 4.1.2) may not catch up, for instance, if the large pose estimation delay (*i.e.*, time between when pose changes and when estimation changes) is large compared to moving speed. In this case, the radio may have to fall back to the 802.11ad standard. However, for the VR system, since pose information feedback is also used for graphics rendering, we believe the pose estimation delay is sufficiently small for Pia to operate correctly under the VR-motion speed. We will have more exploration in the future work once the hardware grants us a better control.

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

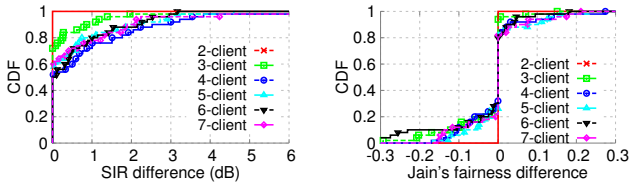


Figure 25: (a) SIR difference (b) Fairness difference between SLR-based and SIR-based assignment.

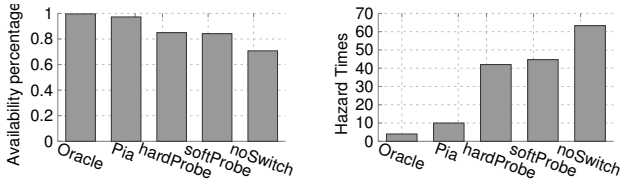


Figure 26: (a) Link availability (b) Hazard times of five AP switching schemes.

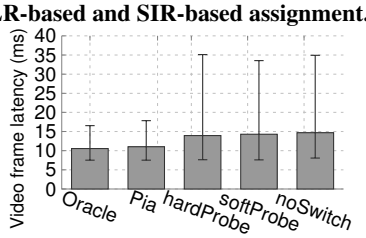


Figure 27: Video frame latency of five AP switching schemes.

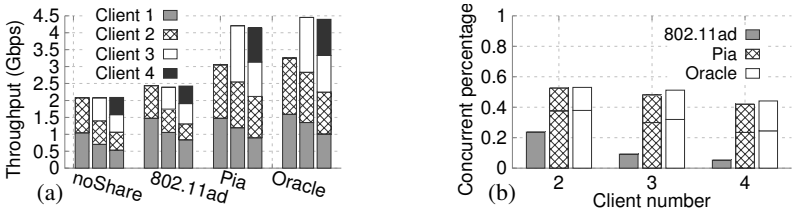


Figure 28: (a) Throughput and (b) chance of concurrency by four spatial sharing schemes.

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 [65]. Pia performs closely to the oracle, with 1.47× and 2× gain over *noShare* 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. A^3 [62] can estimate 3D orientation $<5^\circ$ 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 forwarders 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 (Sec. 6.1). 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

[1] C. Dehos, J. L. González, A. De Domenico, D. Kténas, and L. Dussopt, "Millimeter-Wave Access and Backhauling: The Solution to the Exponential Data Traffic Increase in 5G Mobile Communications Systems?" *IEEE Communications Magazine*, 2014.

[2] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Cutting the Cord in Virtual Reality," in *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)*, 2016.

[3] B. L. Dang, M. G. Larrode, R. V. Prasad, I. Niemegeers, and A. Koonen, "Radio-Over-Fiber Based Architecture for Seamless Wireless Indoor Communication in the 60GHz Band," *Computer communications*, 2007.

[4] "Miracast," http://www.wi-fi.org/wi-fi-certified-miracast/.

[5] IEEE Standards Association, "IEEE Standards 802.11ad-2012: Enhancements for Very High Throughput in the 60 GHz Band," 2012.

[6] "IEEE Standards 802.15.3c," 2009.

[7] "IEEE 802.11ay Study Group," http://www.ieee802.org/11/Reports/tgay_update.htm, 2016.

[8] TPCAST, "HTC VIVE Wireless Upgrade Kit," http://www.tpcast.cn/h_en/index.html, 2016.

[9] S. K. Saha, V. V. Vira, A. Garg, and D. Koutsonikolas, "60 GHz Multi-Gigabit Indoor WLANs: Dream or Reality?" *arXiv preprint arXiv:1509.04274*, 2015.

[10] 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 ACM International Workshop on mmWave Communications: From Circuits to Networks (mmCom)*, 2010.

[11] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi, "Holoportation: Virtual 3D Teleportation in Real-time," in *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)*, 2016.

[12] V. Pichapati, H. Kowshik, A. P. Subramanian, R. Kokku, and M. Chetlur, "Location Assisted Handoffs in Dense Cellular Networks," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2014.

[13] I. Stojmenovic, *Handbook of Wireless Networks and Mobile Computing*. John Wiley & Sons, 2003.

[14] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved Access Point Selection," in *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM, 2006.

[15] W.-S. Soh and H. S. Kim, "QoS Provisioning in Cellular Networks Based on Mobility Prediction Techniques," *IEEE Communications Magazine*, vol. 41, no. 1, 2003.

[16] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. Halvorsen, "Mobile Video Streaming Using Location-based Network Prediction and Transparent Handover," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2011.

[17] HTC, "Vive," https://www.vive.com/us/, 2016.

[18] Google, "Tango," https://get.google.com/tango/, 2014.

[19] Lenovo, "Phab 2 Pro With Tango," http://shop.lenovo.com/us/en/tango/, 2016.

[20] O. Bazan and M. Jaseemuddin, "A Survey On MAC Protocols for Wireless Adhoc Networks with Beamforming Antennas," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, 2012.

[21] Qualcomm, "Qualcomm 802.11ad Products to Lead the Way for Multi-band Wi-Fi Ecosystem," https://www.qualcomm.com/news/releases/2016/01/05/qualcomm-80211ad-products-lead-way-multi-band-wi-fi-ecosystem, 2016.

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

"OpenMili: A 60 GHz Software Radio Platform with a Reconfigurable Phased-Array Antenna," in *Proceedings of ACM MobiCom*, 2016.

[23] C. Rowell and E. Y. Lam, "Mobile-phone antenna design," *IEEE Antennas and Propagation Magazine*, 2012.

[24] G. T. Tapan Pattnayak, "Antenna Design and RF Layout Guidelines," Cypress Semiconductor, 2016.

[25] T. Nitsche, G. Bielsa, I. Tejado, A. Loch, and J. Widmer, "Boon and Bane of 60 GHz Networks: Practical Insights Into Beamforming, Interference, and Frame Level Operation," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. ACM, 2015.

[26] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao, and H. Zheng, "Demystifying 60GHz Outdoor Picocells," in *Proceedings of ACM MobiCom*, 2014.

[27] T. Nitsche, C. Cordeiro, A. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "Ieee 802.11 ad: Directional 60 ghz communication for multi-gbps wi-fi."

[28] Wikipedia, "Rotation Matrix in 3D," https://en.wikipedia.org/wiki/Rotation_matrix, 2016.

[29] Wikipedia, "Four-Quadrant Inverse Tangent," https://en.wikipedia.org/wiki/Atan2.

[30] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

[31] A. Jafari, J. Sarrazin, D. Lautru, A. Benlarbi-Delai, L. Petrillo, and P. De Doncker, "NLOS Influence on 60 GHz Indoor Localization Based on a New TDOA Extraction Approach," in *Microwave Conference (EuMC), 2013 European*. IEEE, 2013.

[32] M. J. Todd and E. A. Yıldırım, "On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids," *Discrete Applied Mathematics*, 2007.

[33] Wikipedia, "Friis Transmission Equation," https://en.wikipedia.org/wiki/Friis_transmission_equation, 2016.

[34] B. Langen, G. Lober, and W. Herzig, "Reflection and transmission behaviour of building materials at 60 ghz," in *Personal, Indoor and Mobile Radio Communications, 1994. Wireless Networks-Catching the Mobile Future., 5th IEEE International Symposium on*. IEEE, 1994.

[35] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling," in *Proceedings of ACM SIGMETRICS*, 2015.

[36] T. Wei and X. Zhang, "Facilitating Robust 60 GHz Network Deployment By Sensing Ambient Reflectors," in *USENIX NSDI*, 2017.

[37] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng, "Reusing 60GHz Radios for Mobile Radar Imaging," in *Proc. of ACM MobiCom*, 2015.

[38] Wikipedia, "Geodesic Grid," https://en.wikipedia.org/wiki/Geodesic_grid.

[39] J. Wang, Z. Lan, C.-S. Sum, C.-W. Pyo, J. Gao, T. Baykas, A. Rahman, R. Funada, F. Kojima, I. Lakkis *et al.*, "Beamforming Codebook Design and Performance Evaluation for 60GHz Wideband WPANs," in *IEEE Trans. on Vehicular Technology*, 2009.

[40] E. Candes and J. Romberg, "L1-Magic: Recovery of Sparse Signals via Convex Programming," *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, 2005.

[41] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[42] A. Khafaji, R. Saadane, J. El Abbadi, and M. Belkasmi, "Ray Tracing Technique based 60 GHz Band Propagation Modeling and Influence of People Shadowing," *IJECSE*, 2008.

[43] Degli-Esposti and etc., "Ray-Tracing-Based mm-Wave Beamforming Assessment," *IEEE Access*, 2014.

[44] E. De Groot, T. Bose, C. Cooper, and M. Kruse, "Remote Transmitter Tracking with Raytraced Fingerprint Database," in *IEEE MILCOM*, 2014.

[45] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-Mile Navigation Using Smartphones," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015.

[46] M. Werner, M. Kessel, and C. Marouane, "Indoor Positioning Using Smartphone Camera," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*. IEEE, 2011.

[47] W. Kang and Y. Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization," *IEEE Sensors journal*, 2015.

[48] W. Kang, S. Nam, Y. Han, and S. Lee, "Improved Heading Estimation for Smartphone-Based Indoor Positioning Systems," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*. IEEE, 2012.

[49] Kvalo, "Linux wil6210 Driver," http://git.kernel.org/pub/scm/linux/kernel/git/kvalo/ath.git/, 2016.

[50] Eberhard GÃijlch, "Investigations on Google Tango Development Kit for Personal Indoor Mapping," https://agile-online.org/conference_paper/cds/agile_2016/posters/102_Paper_in_PDF.pdf, 2016.

[51] M. Costi and R. Dabrowski, "Mobile motion capture," 2014.

[52] "Google Wants to Use Tango Tech for VR, But Admits Current Dev Kits ArenâĂŹt Optimized," http://www.roadtovr.com/google-wants-to-use-tango-tech-for-vr-but-admits-current-dev-kits-arent-optimized/, 2015.

[53] "Virtual STA and AP Interfaces with ath5k, ath9k and ath10k," http://www.candelatech.com/vsta.php, 2014.

[54] "mac80211 Multiple Virtual Interface (vif) Support," https://wireless.wiki.kernel.org/en/users/documentation/iw/vif, 2015.

[55] R. Chandra and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*. IEEE, 2004.

[56] S. Kandula, K. C.-J. Lin, T. Badirkhanli, and D. Katabi, "FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput," in *USENIX NSDI*, 2008.

[57] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating Mobile 60 GHz Networks via Model-driven Beam Steering," in *Proc. of IEEE INFOCOM*, 2016.

[58] iPerf, "The Ultimate Speed Test Tool for TCP, UDP and SCTP," https://iperf.fr/, 2015.

[59] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level Localization With a Single WiFi Access Point," in *Proceedings of Usenix Conference on Networked Systems Design and Implementation (NSDI)*, 2016.

[60] C. Zhang and X. Zhang, "LiTell: robust Indoor Localization Using Unmodified Light Fixtures," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016.

[61] S. Zhu and X. Zhang, "Enabling High-Precision Visible Light Localization in Today's Buildings," in *Proc. of ACM MobiSys*, 2017.

[62] P. Zhou, M. Li, and G. Shen, "Use It Free: Instantly Knowing Your Phone Attitude," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014.

[63] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-Navi: Self-Deployable Indoor Navigation System," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014.

[64] R. Jain, D.-M. Chiu, and W. R. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource*

Allocation in Shared Computer System. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

[65] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "DIRC: Increasing Indoor Wireless Capacity Using Directional Antennas," *ACM SIGCOMM Computer Communication Review*, 2009.

[66] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling Fine-Grained Indoor Localization via Smartphone," in *MobiSys*. ACM, 2013.

[67] Doc-Ok.org, "Lighthouse Tracking Examined," http://doc-ok.org/?p=1478, 2016.

[68] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: Enabling Robust 60 GHz Links Under Blockage," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016.

[69] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "WiFi-Assisted 60 GHz Wireless Networks," in *Proc. of ACM MobiCom*, 2017.

[70] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden, "Improving Wireless Network Performance Using Sensor Hints," in *Proc. of USENIX NSDI*, 2011.

[71] A. Amiri Sani, L. Zhong, and A. Sabharwal, "Directional Antenna Diversity for Mobile Devices: Characterizations and Solutions," in *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*, 2010.

[72] Z. Yang, P. H. Pathak, Y. Zeng, and P. Mohapatra, "Sensor-Assisted Codebook-Based Beamforming for Mobility Management in 60 Ghz WLANs," in *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2015.

[73] R. Suleesathira and S. Kunarak, "Neural Network Handoff in Shadow-Rayleigh Fading," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. IEEE, 2005.

[74] M. S. Dang, A. Prakash, D. K. Anvekar, D. Kapoor, and R. Shorey, "Fuzzy Logic Based Handoff in Wireless Networks," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 3. IEEE, 2000.

[75] K. D. Wong and D. C. Cox, "A Pattern Recognition System for Handoff Algorithms," *IEEE Journal on selected areas in communications*, 2000.

[76] G. Athanasiou, P. C. Weeraddana, C. Fischione, and L. Tassiulas, "Optimizing client association for load balancing and fairness in millimeter-wave wireless networks," *IEEE/ACM Transactions on Networking*, 2015.

[77] M. Park, P. Gopalakrishnan, and R. Roberts, "Interference Mitigation Techniques in 60 GHz Wireless Networks," *IEEE Communications Magazine*, 2009.