



# Enabling Ubiquitous Wi-Fi Sensing with Beamforming Reports

Chenhao Wu\*

The Chinese University of Hong Kong  
Hong Kong SAR  
chenhaowu@link.cuhk.edu.hk

Jun Huang

City University of Hong Kong  
Hong Kong SAR  
jun.huang@cityu.edu.hk

Xuan Huang\*

The Chinese University of Hong Kong  
Hong Kong SAR  
xuanhuang@link.cuhk.edu.hk

Guoliang Xing†

The Chinese University of Hong Kong  
Hong Kong SAR  
glxing@cuhk.edu.hk

## Abstract

Wi-Fi sensing systems leverage wireless signals from widely deployed Wi-Fi devices to realize sensing for a broad range of applications. However, current Wi-Fi sensing systems heavily rely on the channel state information (CSI) to learn the signal propagation characteristics, while the availability of CSI is highly dependent on specific Wi-Fi chipsets. Through a city-scale measurement, we discover that the availability of CSI is extremely limited in operational Wi-Fi devices. In this work, we propose a new wireless sensing system called BeamSense that exploits the compressed beamforming reports (CBR). Due to the extensive support of transmit beamforming in operational Wi-Fi devices, CBR is commonly accessible and hence enables a ubiquitous sensing capability. BeamSense adopts a novel multi-path estimation algorithm that can efficiently and accurately map bidirectional CBR to a multi-path channel based on intrinsic fingerprints. We implement BeamSense on several prevalent models of Wi-Fi devices and evaluated its performance with microbenchmarks and three representative Wi-Fi sensing applications. The results show that BeamSense is capable of enabling existing CSI-based sensing algorithms to work with CBR with high sensing accuracy and improved generalizability.

## CCS Concepts

- Networks → Network protocols; • Human-centered computing → Ubiquitous and mobile computing; • Computer systems organization → Embedded and cyber-physical systems.

## ACM Reference Format:

Chenhao Wu, Xuan Huang, Jun Huang, and Guoliang Xing. 2023. Enabling Ubiquitous Wi-Fi Sensing with Beamforming Reports. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10–14, 2023, New York, NY, USA*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3603269.3604817>

\*Co-primary student authors.

†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

ACM SIGCOMM '23, September 10, 2023, New York, NY, USA

© 2023 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 979-8-4007-0236-5/23/09...\$15.00  
<https://doi.org/10.1145/3603269.3604817>

## 1 Introduction

Recent years have witnessed extensive efforts on developing Wi-Fi sensing systems [43, 54]. Leveraging Wi-Fi signals, these systems are envisioned to enable ubiquitous sensing for a broad range of applications, such as indoor positioning [28, 53], human activity recognition [33, 58], and identification [51, 57].

However, to learn channel properties, current Wi-Fi sensing systems have heavily relied on the channel state information (CSI), which presents a major barrier to their wide deployment on existing Wi-Fi infrastructures and off-the-shelf devices. Specifically, since 802.11 standards do not require the PHY layer to report CSI [32], CSI extraction is thus highly dependent on specific Wi-Fi chipsets and firmware support. Due to the proprietary design of Wi-Fi chips, the availability of such support on commodity Wi-Fi network interface cards (NICs) has been unknown except for only three chip families [19, 20, 52]. In particular, our large-scale measurements conducted on 38,529 operational Wi-Fi devices show that only 6% of Wi-Fi devices may support CSI extraction, which demonstrates a significant gap between the vision of ubiquitous Wi-Fi sensing and the lack of CSI support on the majority of existing Wi-Fi devices.

To alleviate the reliance of Wi-Fi sensing on CSI, recent studies have proposed to utilize *compressed beamforming report (CBR)*, an 802.11 management frame that carries essential channel information for performing beamforming. In contrast to CSI, CBR frames can be easily obtained by sniffing Wi-Fi traffic. However, for reducing extra communication overhead, CBR carries only partial and compressed information about the channel state, presenting significant challenges for sensing. Although there exist several CBR-based sensing systems, they either directly estimate channel properties from CBR or rely on machine learning (ML) models to map CBR for detecting specific events [22, 24, 34, 35]. These limitations result in severely reduced accuracy, and often require prohibitive training overhead in a variety of advanced sensing applications.

In this paper, we propose *BeamSense*, the first system that enables accurate and generalized Wi-Fi sensing with CBR. Fundamentally different from previous CBR-based systems that are only capable of utilizing limited information of CBR, BeamSense can accurately reconstruct CSI from CBR. This allows seamless migration of a broad range of CSI-reliant sensing applications onto off-the-shelf devices without compromising their performance. Therefore, BeamSense enables a ubiquitous sensing capability using the prevalent Wi-Fi infrastructures without special chipsets and firmware supports.

In designing BeamSense, we address two key challenges. First, because CBR integrates a transformed set of channel factors, classic

multi-path models and signal processing algorithms cannot be readily employed to analyze CBR. Second, because deriving CBR from CSI is an irreversible transformation, recovering channel properties from CBR through conventional methods, such as the exhaustive search for path parameters, can be prohibitively expensive. To address these challenges, our key approach is three-fold. We first model the transformation from signal propagation characteristics to CBR. We then define a maximum-likelihood estimation problem to find the multi-path channel that can reproduce the CBR information. To solve this problem, we devise a computationally efficient likelihood evaluation method based on the similarity between sub-channel structures and integrate it into an efficient multi-path estimation algorithm. Our approach is capable of accurately recovering the multi-path channel in real time, enabling CSI-based sensing applications to run seamlessly with CBR.

We evaluate BeamSense using six commodity Wi-Fi NICs of different generations (802.11n/ac/ax) manufactured by five major vendors. We conduct extensive experiments on three controlled testbeds and one large-scale campus Wi-Fi. Our evaluation shows that BeamSense can recover multi-path properties (e.g., AoA, AoD, and ToF) in real time with a high level of accuracy similar to that achieved by CSI-reliant sensing systems. Moreover, BeamSense outperforms the state-of-the-art CSI-based sensing systems which are reliant on CSI tools of legacy Wi-Fi NICs with lower bandwidth and fewer antennas. We further integrate BeamSense with three representative Wi-Fi sensing applications including device localization [15], passive object tracking [36], and sign language recognition [33]. Our evaluation shows that BeamSense enables these applications to work with CBR while achieving high sensing accuracy. In particular, BeamSense can effectively depict the events of interest and significantly improve the accuracy and cross-domain generality compared to the learning-based methods that directly use raw CBR for sensing.

## 2 Related Work

**CSI-based Wi-Fi sensing.** Wireless sensing systems based on radio measurements from widely-deployed commodity Wi-Fi devices have been extensively studied. Early research leverages the Received Signal Strength Indicator (RSSI) as a feature of power to achieve simple fingerprinting in indoor localization [11, 55]. However, due to the fickle and coarse-grained nature, RSSI suffers from severe performance degradation in complex settings [54]. Some commercial devices provide the software interface to obtain high-resolution CSI measurements. As a result, the last decade has witnessed numerous works to leverage CSI measurements for various sensing tasks, including indoor localization [28, 36, 45, 47, 53], human activity recognition [10, 18, 37, 49, 56, 58] and authentication/identification [51, 57]. However, as shown in this paper, CSI is not widely available among deployed commodity Wi-Fi devices, which severely limits the ubiquity of CSI-based Wi-Fi sensing applications in real-world settings. Due to the increasing interest in WLAN sensing, an 802.11 task group [16] is currently standardizing the CSI extraction capability in future premium Wi-Fi devices. However, enabling CSI extraction in accordance with the new standard likely requires new chip designs.

**CBR-based Wi-Fi sensing.** In contrast to CSI, CBR is a more ubiquitous source of sensing data thanks to the growing popularity of

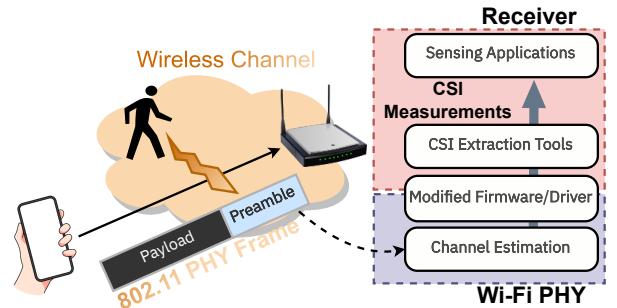


Figure 1: Scenario of CSI-based Wi-Fi Sensing

TX beamforming support on commodity Wi-Fi devices. To perform Wi-Fi sensing without CSI, recent studies have proposed CBR-based sensing methods, which can be divided into two categories.

*Direct CBR-based sensing methods* attempt to detect events directly from CBR. Several recent works take advantage of the observation that periodic changes of channel state may result in specific patterns in CBR variation, allowing to measure the frequency of recurrent events such as respiration rate [24–26]. Other works propose to localize devices by directly computing AoA and AoD based on CBR [22, 25]. However, because CBR contains only partial and compressed channel information, such a CBR-based direct sensing approach suffers significantly reduced accuracy and cannot support advanced sensing tasks such as gesture recognition.

*ML-based sensing methods* rely on training ML models to map CBR to a device location or a sensing event, such as human presence and count [27, 34, 35, 41, 42]. However, such sensing models are highly application-specific, leading to poor generalizability. In particular, because CBR is dependent on a complex set of channel factors, the sensing model must be re-designed and re-trained for each application and deployment scenario, resulting in prohibitive overhead.

**Multi-path estimation.** Wireless channels are known to have sparse structures, where the signals that propagate through several dominant paths decide the received channel states. This characteristic is leveraged in indoor localizations [15, 36, 53], reducing signal blockages [40], eliminating channel feedbacks [12, 46], combating high mobility scenarios [29, 48], and designing intelligent reflecting interfaces [30]. In particular, estimating the multi-path parameters from channel information is an essential step in these problems. However, all existing work requires the knowledge of original CSI in complete or partial frequency domains of various wireless technologies (Wi-Fi/Cellular/mmWave). To the best of our knowledge, there is no existing work to estimate multi-path channels from the partial singular value decomposition results of the channel.

## 3 Background

**Wi-Fi sensing and CSI.** Figure 1 shows a typical setup of a Wi-Fi sensing system. At a high level, Wi-Fi sensing aims to localize a Wi-Fi device or infer the activities of nearby objects by learning how signals propagate from a transmitter to a receiver. To this end, existing Wi-Fi sensing systems rely on the CSI measurements which depict how the preamble of a received Wi-Fi packet is distorted by a wireless channel. Specifically, the CSI of an  $M_{\text{TX}} \times M_{\text{RX}}$  MIMO system with  $K$  subcarriers is a collection of  $K M_{\text{RX}} \times M_{\text{TX}}$  complex

matrices denoted as  $\mathcal{H} \in \mathbb{C}^{K \times M_{\text{RX}} \times M_{\text{TX}}}$ , where the element at  $(i, j)$  of the  $k$ -th matrix describes how the amplitude and phase of a signal in subcarrier  $k$  change when the signal propagates from the  $i$ -th antenna of the transmitter to the  $j$ -th antenna of the receiver. Typically, CSI is calculated in the Wi-Fi chipset for the receiver circuit to demodulate signals. Reporting CSI to the upper layer is not mandatory in the 802.11 standards.

**Compressed beamforming report.** Transmit (TX) beamforming allows the transmitter to exploit spatial diversity by steering the signals in a specific direction. To enable TX beamforming, the transmitter (beamformer) relies on channel information measured at the receiver (beamformee) to learn the steering parameters. However, exchanging full CSI can be bandwidth-consuming, especially for today's Wi-Fi systems with multiple antennas and a large number of subcarriers. The compressed beamforming report (CBR) of 802.11ac is a transformation of CSI, which consists of essential information for performing beamforming but requires significantly lower bandwidth to exchange [9]. Mathematically, an 802.11ac CBR has two components.

**Steering Matrices.** For each subcarrier, the spatial signals are pre-coded to align with a set of the orthogonal basis of the channel matrix measured at the subcarrier. Specifically, for  $k$ -th subcarrier, the steering matrix  $\mathbf{V}_k \in \mathbb{C}^{M_{\text{Bf ee}} \times M_{\text{Bf er}}}$  is right singular vectors from Singular Value Decomposition (SVD) on  $\mathbf{H}_k \in \mathbb{C}^{M_{\text{Bf ee}} \times M_{\text{Bf er}}}$ :

$$\mathbf{H}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^\dagger \quad (1)^1$$

where each column  $\mathbf{v}_{k,i}$  is a steering vector used for adding phase shift on array elements. In general, the steering vectors are invariant to arbitrary phase offset, i.e.,  $\forall d \in [0, 2\pi]$ ,  $e^{j2\pi d} \mathbf{v}_{k,i}$  and  $\mathbf{v}_{k,i}$  have equivalent effects in beamforming. In 802.11, this freedom is compressed with Givens Rotation, where the last row of captured steering matrix  $\tilde{\mathbf{V}}_k$  is always real [9]. The compression is equivalent to multiplying  $\mathbf{V}_k$  with a unitary diagonal matrix  $\mathbf{D}_k$  whose diagonal elements are the column-wise phase shifts to clear the phase of the last row in  $\mathbf{V}_k$ :

$$\tilde{\mathbf{V}}_k = \mathbf{V}_k \mathbf{D}_k \quad (2)$$

**Averaged SNR (ASNR).** To annotate the quality of selected beams, for each spatial subchannel, beamformee averages the estimated SNR over all subcarriers as Eq. 3, and reports to the beamformer<sup>2</sup>:

$$\gamma_i = \frac{1}{K} \sum_k^K 10 \log_{10} \frac{P_{\text{TX}} \cdot \lambda_{k,i}^2}{P_N} \quad i = 1, \dots, M_{\text{Bf er}} \quad (3)$$

where  $\lambda_{k,i}$  is the  $i$ -th singular value in the diagonal matrix  $\Sigma_k$  of Eq. 1,  $P_{\text{TX}}$  is the TX power and  $P_N$  is the measured noise power.

In 802.11ac, a beamformer and a beamformee exchange CBR by following a *channel sounding* protocol. As shown in Figure 2(a), the channel sounding protocol initiates with a control frame *Null Data Packet Announcement* (NDPA) from a beamformer station (STA), where another STA is selected as the beamformee to receive the subsequent NDP frame. NDP is a sounding packet, only containing a standalone frame preamble. Upon measuring the channel states

<sup>1</sup>In this paper, we use  $(\cdot)^T$ ,  $(\cdot)^\dagger$ , and  $(\cdot)^\ddagger$  to denote transpose, conjugate, and conjugate transpose.

<sup>2</sup>Despite Multi-user (MU) CBR including an extra data field for per-subcarrier SNR estimation, our measurement found Single-user (SU) beamforming capability is more ubiquitously supported, thus our work focuses on the information SU CBR carries.

from NDP, the beamformee generates beamforming reports and sends them back to the beamformer.

## 4 Motivation and Objective

**Limited availability of CSI.** Our work is motivated by the limited availability of CSI on off-the-shelf Wi-Fi devices. Specifically, to date, CSI has been available on only three Wi-Fi chipset families [19, 20, 52], which severely hinders the wide deployment and adoption of Wi-Fi sensing applications.

Moreover, in today's Wi-Fi interface cards, the availability of CSI not only depends on chipset design but also requires significant efforts to reverse-engineer and modify chip firmware.

- First, since the 802.11 standards do not require the physical layer to report CSI, commodity Wi-Fi interface cards typically calculate CSI inside the chipset without providing a data path for sending CSI to the host device. To date, only three Wi-Fi chip families, *ath9k*, *Intel5300*, and *nexmon*, are known to have built-in CSI data paths for debugging purposes. Unfortunately, the availability of a similar data path is unknown in the majority of Wi-Fi interface cards due to the proprietary nature of the chipset design.
- Second, even if the CSI data path is available in a Wi-Fi chip, enabling it for Wi-Fi sensing would require significant engineering efforts. For example, existing CSI extraction tools need to first put the Wi-Fi chip into debug mode using chip-specific commands and then read CSI from specific memory addresses. Since chipset designs vary, enabling ubiquitous Wi-Fi sensing would require chip-specific reverse engineering and firmware modification, which can be prohibitively expensive.

**Generalized Wi-Fi sensing with CBR.** Contrary to CSI, CBR frames can be easily obtained via wireless traffic sniffing using commodity Wi-Fi devices<sup>3</sup>, making CBR-based sensing a promising paradigm.

Our key objective is to enable ubiquitous Wi-Fi sensing by recovering CSI from CBR, as the multi-path information in CSI is the foundation of a diverse set of sensing systems including localization, tracking, and human activity recognition. Our objective fundamentally differs from existing CBR-based approaches, which perform sensing by directly applying machine learning or sensing algorithms on CBR, yielding poor performance and limited generalizability due to the partial and compressed channel information in CBR. By addressing these limitations, we aim not only to achieve accurate and generalized CBR-based Wi-Fi sensing, but also to allow the broad range of existing CSI-reliant sensing systems to migrate onto the already-prevalent 802.11ac-enabled devices.

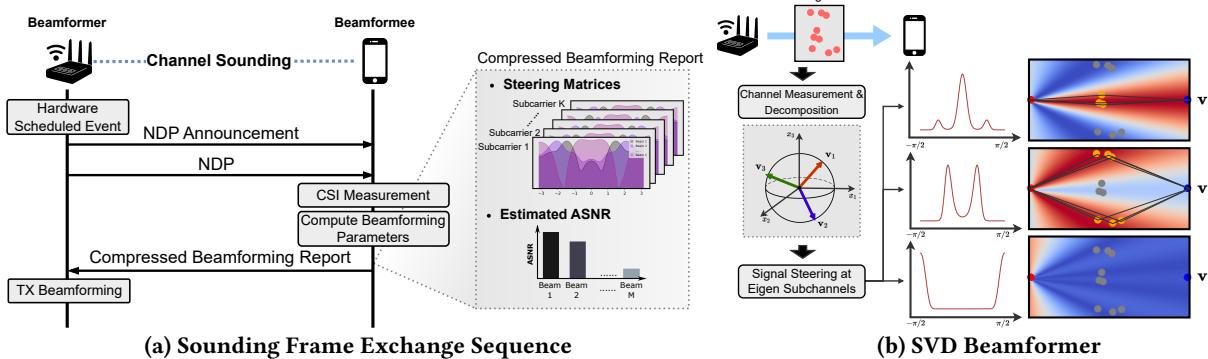
## 5 Ubiquity of Wi-Fi Sensing: A Measurement Study

In this section, we present a measurement study to understand the ubiquity of CSI (§5.1), and the TX beamforming (TXBF) support in the current infrastructure (§5.2). Lastly, we conclude our key findings (§5.3).

### 5.1 Can CSI Enable Ubiquitous Sensing?

We answer this question by investigating the deployable rate of CSI extraction tools on commodity devices. The measurement examined

<sup>3</sup>The triggering of channel sounding comprises a normal control frame and an 802.11 preamble, which can be accomplished via pure layer 2 operations (e.g., via packet injection and emulation [14, 23]) even in the absence of beamformer capability.



*Figure 2: 802.11 Channel Sounding and Transmit Beamforming.*

**Table 1: CSI and beamforming capabilities in a city-scale measurement.**

		AP	STA
CSI Tool	# Intel5300	0% (0)	$\leq 0.004\%$ (86)
	# ath9k	$\leq 4.7\%$ (876)	$\leq 1.5\%$ (292)
	# nexmon	0% (0)	$\leq 5.4\%$ (1,066)
	Subtotal	$\leq 4.7\%$ (876)	$\leq 7.3\%$ (1,444)
Sensing w/ CBR		57.1% (10,712)	$\geq 45.5\%^*$ (3,592)
Total		18,745	19,784

over 38,529 Wi-Fi devices (18,745 operational AP and 19,784 STA) that have been deployed in a city.

**Methodology.** To examine thousands of deployed devices efficiently, we conduct the measurement by analyzing the public Wi-Fi traffic. We set up a laptop operating in monitor mode and walk around the city to collect Wi-Fi frames. The same route is repeated to iterate over all legitimate WLAN channels on 2.4GHz and 5GHz. In total, we captured 1,360,713 frames (elapsing 610 minutes) for analysis.

Examining the compliance with any CSI extraction tools of a device in the packet trace is not easy, as this capability is dependent on the product model which is not encapsulated in public 802.11 frames. To address this issue, we exploit Wi-Fi chipset knowledge provided by the public device tree<sup>4</sup> and publicly available chipset specifications. We then designed a two-step filtering pipeline, which analyzes collected packet traces to exclude the devices that do not meet the conditions to install any existing CSI extraction tools.

- **Vendor-based filtering.** First, we exclude all Wi-Fi devices from vendors which do not employ the three chipset families that support CSI tools. To this end, we first shortlist all vendors that should be excluded by checking the public device tree, and then identify all Wi-Fi devices from these vendors based on the OUI (Organization Vendor Identifier) of packet MAC addresses.
  - **Radio capability-based filtering.** Then, we further filter remaining devices by exploiting radio capabilities for finer-grained chipset fingerprinting. We focus on four capabilities, namely *max.standard*, *band*, *max.MCS*, and *beamforming*. These capabilities of a specific Wi-Fi device can be learned based on action-specific frames like *NDPA* and *QoS Data*, the data carried by

beacon frames (if the device is an access point), and the device's response to frames modulated with specific MCS. It is worth noting that capability-based chipset fingerprinting cannot exclude all devices that do not support CSI. As a result, our estimation of CSI tools' deployable rate is optimistic.

**Result.** The upper part of Table 1 presents the optimistic deployable rate of existing CSI extraction tools. Because the radio capabilities of APs are more reliably understood (via beacon frames), we separate the results by the operation mode (*i.e.* AP or STA) of devices. On the AP side, we are only able to shortlist 876 devices (among 18,745 OUI-Valid devices) that fall in the device tree of ath9k and align with the official specifications, which counts for 4.7% of the total AP devices. The STA result shows a little higher deployable rate, which counts for 7.3% of all STA devices. In particular, as the chipset family supported by nexmon is adopted by several popular smartphone models (*e.g.*, Apple iPhone6 and Samsung Galaxy Series), our pipeline shortlists 1,066 devices that can possibly support nexmon. Nonetheless, the aggregate deployable rate is still less than 6%.

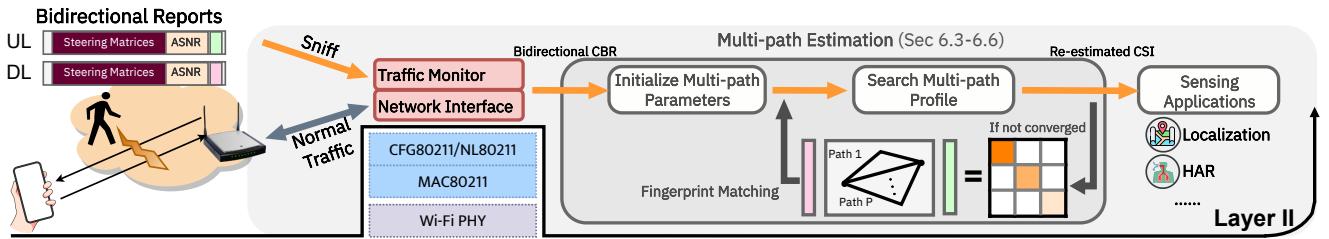
## 5.2 TX Beamforming Support

As depicted in Section 4, *sensing with CBR* requires the devices to be able to respond with the channel sounding announcements, which is a basic component in the TX beamforming subsystem. Therefore, the ubiquity of the CBR sensing scheme can be understood by investigating the TX beamforming support among deployed Wi-Fi devices.

**Methodology.** From the same dataset in §5.1, we crawl the devices that advertised beamforming-related capabilities in their *Beacon* frames, as well as all the sender and receiver devices of frames used in channel sounding.

**Result.** We discovered 10,712 operational APs that advertise SU/MU-Beamforming capabilities in their emitted beacon frames, which counts for 57.1% of the total AP devices in the packet trace. These AP devices are capable of reporting CBR when receiving channel sounding announcements from a connected device. Moreover, nearly 80% of these AP devices can perform as beamformers, which implies they can automatically provoke channel sounding to connected STA. Specifically, we captured NDPA frames sent from 3,215 devices among them. On the other hand, among 7,891 STA devices that have connected to APs with TX beamforming support, we

<sup>4</sup>Public device tree [2, 4, 17] includes most of the commodity models supported by kernel drivers like ath9k and Intel5300.



**Figure 3: Overview of BeamSense:** BeamSense estimates multi-path parameters from sniffed bidirectional CBR frames and feeds the re-estimated channel states to downstream sensing applications. To collect CBR for sensing, BeamSense can either sniff ambient traffic to collect the CBR frames of nearby Wi-Fi links or be deployed on one end of a Wi-Fi link to extract CBR frames from the local Wi-Fi interface working in the promiscuous mode.

found 45.5% (3,592) of STA devices have reported CBR frames to their APs. In total, we identified 14,304 deployed devices that can be used for sensing with CBR.

### 5.3 Findings

The implication of this measurement study is two-folded:

- First, we reveal that even in optimistic estimation, the conventional CSI-based sensing scheme can be deployed on less than 6% of all Wi-Fi devices. We attribute this low penetration rate to the fact that CSI extraction capability is not required by the standards. Moreover, two of the three chipset families, ath9k and Intel5300, are 802.11n chipsets. However, among 18,745 AP devices, we only find 28.7% (5,376) devices that support up to 802.11n.
- On the contrary, our measurement study characterizes a decent installation rate of TX beamforming subsystems in current Wi-Fi deployments. Over half of the deployed devices are capable of participating in TX beamforming and reporting CBR frames. The penetration rate is as high as 57.1% in all operational APs and, particularly, 73.2% in the devices compliant with 802.11ac and newer standards. Bidirectional CBR can be generated from these devices via layer II operations and used for sensing.

## 6 Design of BeamSense

### 6.1 Challenges

Instead of relying on CSI which is only available on a few Wi-Fi chips, we aim to enable generalized Wi-Fi sensing with CBR which is already supported by prevalent 802.11ac-enabled devices. Realizing this goal faces several challenges.

**Significant domain shift in CBR.** As discussed in §3, CBR is derived from the SVD factorization of CSI, and contains only partial and transformed information about signal propagation characteristics. Therefore, CBR cannot be directly used by existing CSI-based Wi-Fi sensing algorithms. To address this issue, a straightforward approach is to re-design and train ML models from the samples of CBR. However, as the SVD is a synthetic result of multiple factors of the channel, significant gaps would exist between the CBRs captured with different environments, objects/activities of interest, and RF configurations. Therefore, simply using CBRs to fingerprint sensing events can be expensive in training and lacks generalizability.

**Mapping CBR to CSI.** Instead of designing sensing algorithms directly based on CBR, recovering CSI from CBR will not only enable generalized Wi-Fi sensing but also allow seamless migrations

of existing CSI-based sensing systems to the majority of Wi-Fi chips. However, it is challenging to map CBR to the original CSI. Since SVD is known to be a one-way factorization and bidirectional CBR only depicts partial SVD results, the original CSI matrix cannot be derived with a closed-form solution or with existing signal processing techniques.

**Computational overhead of multi-path estimation.** In this paper, we leverage the multi-path channel model to understand the interplay between the physical channel and the corresponding bidirectional CBR. Based on the model, a naive approach to obtain a multi-path channel is to apply maximum-likelihood estimation (MLE) to exhaustively search the entire solution space of multi-path parameters, compute the SVD of each solution, and compare the SVD result with bidirectional CBR. However, such a naive approach is highly complex in theory and prohibitively expensive in practice. In particular, the time complexity of SVD is proportional to the cube of the number of antennas, which further scales with the number of subcarriers. Through empirical measurements conducted on a laptop with 3.2GHz CPU, we find that computing the SVD with the widely used LAPACK library takes about 0.5ms for the CSI of a 4×4 MIMO system. Moreover, the solution space has an extraordinarily high dimension. For example, in order to search the space of 6 paths with a step of one degree AoA and AoD and of one nanosecond path delay, there will be  $2^{128}$  possible multi-path profiles to be evaluated. Combined with the non-negligible overhead in evaluating SVD, the overall computation delay is prohibitively high, which can be over days.

### 6.2 Overview of BeamSense

To enable generalized Wi-Fi sensing without relying on special chip and firmware support, we design *BeamSense*, a system that can reconstruct a multi-path channel from bidirectional CBR accurately and efficiently. BeamSense can be deployed as an underpinning layer below existing CSI-based Wi-Fi sensing systems, enabling them to operate seamlessly on CBR and migrate to the majority of existing Wi-Fi infrastructures and off-the-shelf devices where CSI is not available.

To reconstruct a multi-path channel from CBR, we characterize the relationship between signal propagation characteristics and the information carried by CBR, and then devise a computationally efficient algorithm to tackle the key challenges discussed in §6.1. First, rather than factorizing multi-path channels to search for

matched path parameters, BeamSense leverages the key observation that multi-path channels render unique subchannel structures after SVD, which can be exploited as a fingerprint to match path parameters with bidirectional CBR without actually computing SVD. Leveraging this insight, we formulate multi-path estimation based on a computationally efficient MLE. Instead of maximizing the match of SVD results, BeamSense seeks to minimize the error of fingerprint match, which significantly reduces computational overhead. Moreover, we optimize parameter search based on iterative expectation maximization and strategic initialization points optimization, which help BeamSense converge to an accurate estimation with a significantly reduced search overhead, enabling multi-path estimation from CBR in real time.

### 6.3 Multi-path Modeling with CBR

We analyze the relationship between signal propagation characteristics and the information in CBR by modeling a multi-path channel based on uplink and downlink steering matrices. The model enables accurate and efficient multi-path estimation from CBR, as detailed in §6.4.

**From physical paths to CSI.** The CSI of a multi-path channel describes how a signal changes as it propagates along multiple paths from a transmitter to a receiver. Specifically, the CSI of a signal traversing a single path can be expressed as,

$$\mathbf{h}(f_k, \alpha_p, \phi_p, \psi_p, \tau_p) = \alpha_p e^{-j2\pi f_k \tau_p} \mathbf{F}(\phi_p, \psi_p) \quad (4)$$

where  $f_k$  is the transmission frequency,  $\alpha_p$  is the attenuation factor,  $\phi_p$  and  $\psi_p$  are the Angle of Arrival (AoA) and Angle of Departure (AoD), and  $\tau_p$  is a delay incurred by propagation and Doppler effects. For a  $M_{RX} \times M_{TX}$  MIMO system,  $\mathbf{F}(\phi_p, \psi_p) \in \mathbb{C}^{M_{RX} \times M_{TX}}$  is a matrix that represents how signal phases change across antennas.  $\mathbf{F}(\phi, \psi)$  can be expressed as,

$$\mathbf{F}(\phi, \psi) = \vec{\mathbf{a}}(\phi) \otimes \vec{\mathbf{d}}(\psi) \quad (5)$$

where  $\vec{\mathbf{a}}(\phi)$  and  $\vec{\mathbf{d}}(\psi)$  are the array vectors of the transmitter and receiver, depending on the shape of antenna arrays. For instance, for a uniform linear array where adjacent antennas are separated by  $s$ , the array vectors are,

$$\vec{\mathbf{a}}(f, \phi) = [1, e^{-j2\pi f s \sin \phi / c}, \dots, e^{-j2\pi f s M_{RX} \sin \phi / c}]$$

$$\vec{\mathbf{d}}(f, \psi) = [1, e^{-j2\pi f s \sin \psi / c}, \dots, e^{-j2\pi f s M_{TX} \sin \psi / c}]$$

A multi-path channel is the sum of single-path channels, which can be expressed as

$$\hat{\mathbf{H}}_k(\{\hat{\alpha}_p, \hat{\phi}_p, \hat{\psi}_p, \hat{\tau}_p\}_{p=1}^P) = \sum_p \mathbf{h}(f_k, \hat{\alpha}_p, \hat{\phi}_p, \hat{\psi}_p, \hat{\tau}_p).$$

where  $P$  denotes the number of paths.

**From CSI to CBR.** Next, we model a multi-path channel  $\mathbf{H}$  with bidirectional CBR. Without loss of generality, we focus on the downlink channel at a specific subcarrier.

Since both uplink and downlink signals traverse the same physical paths, their channel matrices at the same frequency are the transpose of each other (*i.e.*, swapping index of TX-RX antennas) after normalizing the transmit power. Thus, the corresponding downlink multi-path channel  $\mathbf{H}$  can be decomposed with either the uplink or downlink steering matrices, as shown in Eq.6.

$$\mathbf{H} = \mathbf{U}_{DL} \Sigma \tilde{\mathbf{V}}_{DL}^\dagger = \tilde{\mathbf{V}}_{UL} \Sigma \mathbf{U}_{UL}^T \quad (6)$$

where  $\tilde{\mathbf{V}}_{DL}$  and  $\tilde{\mathbf{V}}_{UL}$  are the steering matrices decoded from downlink and uplink CBR. Since  $\tilde{\mathbf{V}}_{UL,k}$  and  $\mathbf{U}_{DL}$  are both left singular matrix of  $\mathbf{H}$ , there is a column-wise phase shift  $\mathbf{D}$  such that  $\mathbf{U}_{DL} = \tilde{\mathbf{V}}_{UL} \mathbf{D}$ . To this end, the downlink channel can be expressed with bi-directional CBR as:

$$\mathbf{H} = \tilde{\mathbf{V}}_{UL} \mathbf{D} \Sigma \tilde{\mathbf{V}}_{DL}^\dagger \quad (7)$$

Since the subchannel gains  $\Sigma$  and the coordinate rotation matrix  $\mathbf{D}$  are unknown from CBR, the SVD of a multi-path channel cannot be directly reversed. This also prevents the straightforward application of existing multi-path estimators [22, 28, 38, 53] and phase analysis algorithms [45] for directly computing channel properties, resulting in a key barrier to perform Wi-Fi sensing using CBR.

### 6.4 Maximum Likelihood Multi-path Estimation

Because deriving CBR from CSI is an irreversible transformation, BeamSense estimates a multi-path channel from CBR by searching for path parameters that reproduce the bidirectional CBR. To this end, a naive approach would factorize candidate multi-path channels and examine SVD results, which is however prohibitively expensive due to the cost of SVD. Moreover, because CBR is derived from transformed channel factors that are highly dependent on the deployment environment, data-driven approaches, such as leveraging ML models to infer channel properties from CBR, would result in prohibitive training overhead.

To overcome this issue, BeamSense leverages the unique subchannel structures of CBR as a multi-path channel fingerprint and designs a novel multi-path estimation algorithm. We define an MLE problem to maximize fingerprint matching, which does not require expensive SVD and therefore significantly reduces compute overhead.

**Multi-path fingerprint in CBR.** Our key observation is that multi-path channels can be fingerprinted based on the unique subchannel structures after SVD. Specifically, we exploit two subchannel structures.

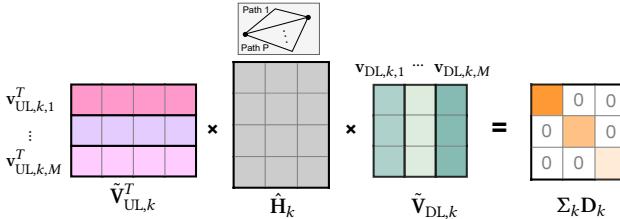
(1) *Null Gains of Non-Diagonal Terms:* Because (DΣ) in Eq. 7 is a diagonal matrix, all non-diagonal elements of  $\tilde{\mathbf{V}}_{UL}^T \mathbf{H} \tilde{\mathbf{V}}_{DL}$  are equal to zero.

(2) *Diagonal Gains Constrained by ASNR:* By Eq. 7, the norm of diagonal terms  $\|x_{ii}\|$  in  $\tilde{\mathbf{V}}_{UL}^T \mathbf{H} \tilde{\mathbf{V}}_{DL}$  are supposed to equal the singular values of  $\mathbf{H}$ , *i.e.*, the subchannel gains. However, the per-subcarrier channel gains are not included in CBR. Therefore, for the  $i$ -th subchannel, we use the  $i$ -th DL ASNR report  $\Upsilon_{DL,i}$  to formulate the constraints of the sum  $\sum_{k=1}^K \|k x_{ii}\|$ . Specifically, we first derive the geometric mean of the  $i$ -th subchannel gains as Eq. 8:

$$\bar{\lambda}_{iG} = (\prod \lambda_{k,i})^{1/K} = 10^{\Upsilon_i/20} \cdot \sqrt{P_N/P_{TX}} \quad (8)$$

We further leverage the characteristics of Wi-Fi signals that the powers across subcarriers must satisfy a low Peak-to-Average-Power ratio as required by signal demodulation [21], which implies  $\bar{\lambda}_i \gg \text{Var}(\lambda_i)$ . Under this constraint, the arithmetic mean of subchannel gains  $\bar{\lambda}_i$  can be approximated by  $\bar{\lambda}_{iG}$  [13]. Finally, the sum of diagonal gains are subject to  $\sum_{k=1}^K \|k x_{ii}\| = K \bar{\lambda}_{iG}$ ,  $\forall i \in [1, M_{TX}]$ .

**Fingerprint matching.** Given a multi-path profile  $\{\alpha_p, \phi_p, \psi_p, \tau_p\}$  and a pair of uplink CBR  $\{\tilde{\mathbf{V}}_{UL,k}, \{\Upsilon_{UL,i}\}\}$  and downlink



**Figure 4: An illustration of fingerprint match where  $T_k = \tilde{V}_{UL,k}^T \hat{H}_k \tilde{V}_{DL,k}$ . The non-diagonal terms are nullified and diagonal terms are constrained by ASNR.**

CBR [ $\{\tilde{V}_{DL,k}\}, \{Y_{DL,i}\}$ ], BeamSense computes a *fingerprint matching matrix*  $T$ , which is expressed as,

$$T = \tilde{V}_{UL}^T \hat{H}(\{\alpha_p, \phi_p, \psi_p, \tau_p\}) \tilde{V}_{DL} \quad (9)$$

To understand why  $T$  characterizes fingerprint match, consider the effect of each product in the SVD factorization. As shown in Figure 2(b), SVD extracts the orthogonal subchannels from the measured channel matrix [44], where  $U$ ,  $\Sigma$ ,  $V$  depict the *output subspace*, *subchannel gains*, and *input subspace*, respectively. The input subspace and the output subspace jointly describe the subchannel structures at a given subcarrier. Each input and output subspace is represented by a set of unitary biorthogonal vectors. The factorization uniquely pairs an input vector in  $V$  with an output vector in  $U$  in a way that for every paired vectors  $u$  and  $v$ , the product  $u^T \cdot H \cdot v$  is the gain of this subchannel  $\lambda$ , and for every unpaired vectors  $u^\perp$  and  $v$ , the product  $(u^\perp)^T \cdot H \cdot v$  is null. Based on this observation, we can exploit  $(u, v)$  to bypass the computation of SVD. As shown in Figure 4, the estimated multi-path channel is *matched* with the actual multi-path channel if all non-diagonal terms are zero and the mean of diagonal terms match the ASNR.

**MLE-based multi-path reconstruction.** For each subcarrier, we construct a fingerprint matching matrix  $T_k$ . The likelihood of a candidate multi-path profile is computed based on fingerprint match, which is characterized using a loss function,

$$\begin{aligned} \mathcal{L}^*(\mathcal{H}) &= \sum_{i=1}^M \sum_{k=1}^K (\|k_{tii}\| - \bar{\lambda}_{iG}) + \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^K \|k_{tij}\| \\ \{\hat{\alpha}_p, \hat{\phi}_p, \hat{\psi}_p, \hat{\tau}_p\} &= \underset{\{\alpha_p, \phi_p, \psi_p, \tau_p\} \in \Omega}{\operatorname{argmin}} \mathcal{L}^*(\mathcal{H}) \end{aligned} \quad (10)$$

where  $\Omega$  is the search space of multi-path profiles.

We note that, although the asymptotic complexity of computing  $T$  is the same as reproducing CBR, the coefficient is much lower. Specifically, the coefficient of computing  $T$  with LAPACK is far below one. In comparison, the coefficient of computing SVD can range from 8 to 20.

## 6.5 Searching Multi-path Parameters

Multi-path channels feature a high-dimensional parameter space, which can be highly expensive to search. For example, a naive approach to this problem is to divide the parameter space into fixed-size grids and then examine all grids, which would result in a prohibitive complexity of  $\Omega_\alpha^P \times \Omega_\phi^P \times \Omega_\psi^P \times \Omega_\tau^P$ . To address this problem, BeamSense employs two optimization strategies, namely iterative searching and seeded initialization.

---

### Algorithm 1: BeamSense Multi-path Estimation

---

```

Input:  $\{\mathbf{V}_{DL,k}\}_{k=1}^K, \{\mathbf{V}_{UL,k}\}_{k=1}^K, \{Y_{DL,i}\}_{i=1}^M$ 
Output:  $\{\hat{\alpha}_p, \hat{\phi}_p, \hat{\psi}_p, \hat{\tau}_p\}_{p=1}^P$ 
1 Initialize starting point of  $\{\alpha_p^0, \phi_p^0, \psi_p^0, \tau_p^0\}_{p=1}^P$ ;
2  $\epsilon$ : Convergence Threshold;
3 Initialize  $g^0 = +\infty$ ,  $t = 1$ ;
4 while converged == false do
5   for each path  $p = 1, 2, \dots, P$  do
6     for each path parameter  $\eta$  in  $(\alpha_p^{t-1}, \phi_p^{t-1}, \psi_p^{t-1}, \tau_p^{t-1})$ 
9       do
10       $\eta_p^t \leftarrow \underset{\eta_p \in \Omega_\eta}{\operatorname{argmin}} \mathcal{L}^*(\eta_p)$ ;
11    end
12  end
13   $g_t \leftarrow \mathcal{L}^*(\{\mathbf{H}_k(\{\alpha_p^t, \phi_p^t, \psi_p^t, \tau_p^t\}_{p=1}^P)\}_{k=1}^K)$ ;
14  if  $\|g_t - g_{t-1}\| < \epsilon$  then
15    converged = true;
16     $\{\hat{\alpha}_p, \hat{\phi}_p, \hat{\psi}_p, \hat{\tau}_p\}_{p=1}^P \leftarrow \{\alpha_p^t, \phi_p^t, \psi_p^t, \tau_p^t\}_{p=1}^P$ ;
17  end
18   $t = t + 1$ ;
19 end

```

---

**Iterative searching.** Instead of searching the parameter space based on grids, BeamSense performs iterative expectation maximization (EM) with the coordinate descent strategy [50]. In each iteration, BeamSense optimizes parameters sequentially. For each parameter, an optimal value is searched while keeping all other parameters fixed. The iteration is terminated when Eq. 10 converges. Although iterative searching does not assure a globally optimal solution, convergence is guaranteed due to the non-increasing objective function. In this way, the number of searched grids is reduced to  $\sum_{p=1}^P (\Omega_\alpha + \Omega_\phi + \Omega_\psi + \Omega_\tau)$ .

**Seeded initialization.** Similar to most non-convex optimization problems, the performance of iterative searching depends on the choice of initialization point. To address this issue, we employ a strategy similar to genetic search. As described in Algorithm 1, to initiate iterative searching, we run optimization on coarse-grained grids and pick the optimal  $N$  grids as initialization points, where we start a new search from each point. Finally, the optimal parameters among the  $N$  searches are used to reconstruct the multi-path channel.

## 6.6 Implementation

This section presents the implementation of BeamSense.

**Software.** After the link establishments, as scheduled by the on-chip timer, two connected devices will initiate channel sounding in turns, and bidirectional CBR frames will show up in the wireless traffic. We deploy BeamSense on one of the devices that runs a concurrent *monitor* interface and use *Scapy* library [31] to read the exchanged frames. Steering matrices and ASNRs are decoded from captured CBR frames in accordance with the 802.11 standards. Before multi-path estimation, we interpolate the bidirectional CBR with a modified *sleep* interpolator [39] to obtain a time-synchronized bidirectional report series. The report series is then

passed to the multi-path estimator implemented with C++, which runs Algorithm 1 to produce estimated CSI series.

**Grid space.** In each iteration of the multi-path estimation, BeamSense finds the path parameters on a grid space that optimizes  $\mathcal{L}^*$  (Line 5-8 of Algorithm 1). The range and resolution of grid space are designed with a tradeoff between accuracy and computation efficiency. We empirically set the searching range of the angular parameters to  $[-\frac{\pi}{2}, \frac{\pi}{2}] rad$  with step size  $0.03rad$ , range of the delay parameters to  $[0, 150] ns$  with step size  $1ns$ , and range of attenuation parameters to  $[0, 1]$  with step size  $1\%$  across all the experiments.

**Eliminating radio chain offsets.** The captured CBR is distorted by the Radio Chain Offsets (RCO), which is caused by the disparity in the antenna chains, and will lead to inaccurate AoA/AoD estimations. With RCO existing at both TX and RX arrays, the channel measurement is distorted by  $\mathbf{H}_{\text{dist}} = \Lambda_{\text{RX}} \mathbf{H} \Lambda_{\text{TX}}$ , where  $\Lambda_{\text{RX}}$  and  $\Lambda_{\text{TX}}$  are diagonal RCO matrices and the diagonal terms denote the phase offsets between antenna elements. We find that RCO is constant over time and frequency domain. Based on this observation, we devise a simple cleaning scheme to eliminate its effect. We first connect the neighboring TX chains and RX chains with coaxial cables and calculate the ideal steering matrices  $\mathbf{V}_{k,\text{ideal}}$  by the known cable lengths. Since RCO incurs column-wise unit phase shift to  $\mathbf{H}_k$ , the captured steering matrix will be distorted by  $\tilde{\mathbf{V}}_k = \Lambda_{\text{Bfer}}^\dagger \mathbf{V}_{k,\text{ideal}}$ . Then, the RCO at the beamformer's array can be estimated by Eq. 11 and used to compensate for CBR captured in the same run.

$$\Lambda_{\text{Bfer}} = \underset{\Lambda \in \Omega_\Lambda}{\operatorname{argmin}} \sum_{k=1}^K \left\| \tilde{\mathbf{V}}_k - \Lambda^\dagger \mathbf{V}_{k,\text{ideal}} \right\|_2 \quad (11)$$

On the other hand, CBR is immune to 3 types of radio errors that are commonly considered in previous Wi-Fi sensing systems, *i.e.*, sampling frequency offset, symbol timing offset, and carrier frequency offset [32]. Because at each subcarrier, these 3 types of errors cause identical phase shifts to all antenna pairs, and therefore the SVD result is invariant.

## 7 Evaluation

This section presents the evaluation of BeamSense in two real-world testbeds. We first examine the performance of BeamSense with several microbenchmarks (§7.2) and then evaluate the performance of three representative Wi-Fi sensing applications with BeamSense estimated channel states (§7.3, §7.4, §7.5).

### 7.1 Methodology

We conduct the experiments in two types of indoor testbeds (as shown in Figure 5):

- **Controlled Testbed:** We conduct the majority of experiments in a controlled testbed, where we use controlled devices to set up all the sensing links. This testbed comprises three different indoor environments. The first is a small meeting room ( $7.2 \times 5.4 m^2$ ) with a desk and some furniture. The second is a large lecture room ( $12 \times 10 m^2$ ) with multiple rows of plastic chairs and a narrow aisle. The last one is a corridor of  $1.8 m$  in width and  $2.5 m$  in height with concrete walls and ceiling. The three different environments correspond to different levels of multi-path conditions for evaluation.
- **Public Infrastructure Testbed:** We also conduct part of our experiments using sensing links established between our devices

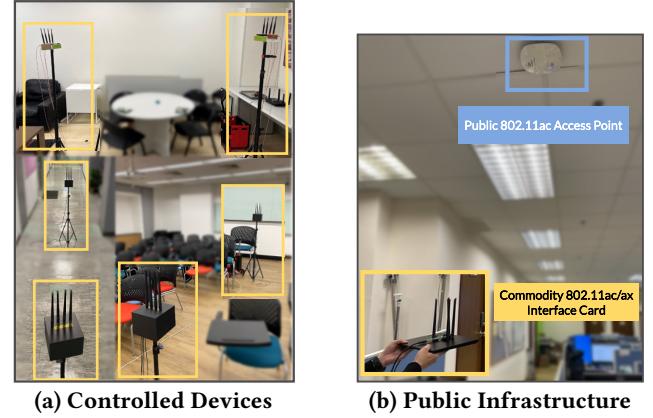


Figure 5: Testbed photos and hardware setup.

and existing AP devices publicly installed in our office ( $35 \times 12 m^2$ ) and the lecture room. The inter-element spacing in the antenna array of public APs is empirically obtained with offline experiments.

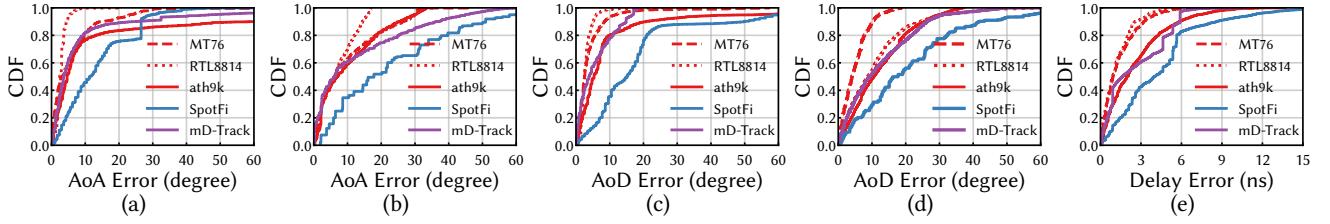
We tested BeamSense using the commodity devices listed in Figure 7, which include five 802.11ac/ax devices with beamformer capability, as well as an 802.11n router that supports *ath9k* CSI extraction tools for comparison with CSI-based algorithms. We assemble our own antenna array with  $2.7 cm$  inter-element separation and use SMA interfaces to connect the RF frontend.

Each Wi-Fi interface card is connected to a laptop that runs *Linux 5.15*, and all routers are configured to run *openwrt* with wired to the laptop. The laptop has 4 CPU cores running at 3.2GHz. We connect devices in a typical AP-STA setup using the userspace applications *hostapd* and *wpa\_supplicant*, operating on the 5GHz band for 802.11ac devices and the 2.4GHz band for 802.11n/ax devices. To enable 802.11ac/ax beamforming, *hostapd* is launched with *[SU-Beamformer]*, *[SU-Beamformee]*, the highest *[Sounding-Dimensions]* and the largest possible bandwidth. We observe that the frequency of the Wi-Fi chipset initiating channel sounding depends on the current network usage. Therefore, we run a pair of *iPerf3* connections in the background to sustain a 1 MBps ambient traffic, which is also compliant with regular usage. The maximum achievable sounding frequency is around 30 Hz. For every single experiment, we evaluate BeamSense with a pair of Wi-Fi devices from the same model, and they are placed in the designated locations with clear LOS. The experiments of this paper does not raise any ethical issues.

### 7.2 Multi-path Estimation Performance

In this section, we evaluate our multi-path estimation performance in the controlled testbed.

**Multi-path estimation accuracy.** To evaluate the accuracy of BeamSense in estimating the multi-path profiles, we place the Wi-Fi transceivers in multiple test locations and use BeamSense to estimate the multi-path profile. We manually set a reflection path using a strong reflector, and focus on the accuracy of estimating the parameters of LOS and reflection path. Because both transceivers and the reflector are static, we obtain the ground truth of parameters based on their actual locations.



**Figure 6:** Estimation accuracy of BeamSense with CBR captured from MT76 and RTL8814, BeamSense with CBR generated from ath9k, and baselines with CSI captured from ath9k. CDF of (a) AoA estimation errors of the direct path; (b) AoA estimation errors of the reflection path; (c) AoD estimation errors of the direct path; (d) AoD estimation errors of the reflection path; (e) relative range estimation errors, for BeamSense, mD-Track, and SpotFi.

	Model	Wi-Fi Chipset	Radio Spec.
1	WNDR4300 [3]	AR9344	2 × 2, 40MHz, 802.11n
2	EDUP-1622 [1]	RTL8814	2 × 2, 80MHz, 802.11ac
3	Mercury UD19H	RTL8814	4 × 3, 80MHz, 802.11ac
4	Xiaomi CR6608 [8]	MT76	2 × 2, 80MHz, 802.11ax
5	RG-EW3200GX [5]	MT76 (MT7915AN)	4 × 4, 80MHz, 802.11ax
6	Tenda AC23 [6]	RTL8814	4 × 4, 80MHz, 802.11ac

**Figure 7: Specifications of tested COTS Wi-Fi Devices.** The first device is legacy 802.11n devices that have provided CSI extraction tools. 2~3 are commodity 802.11ac adapters designed for PC and laptops. 4~6 are commodity 802.11ac/ax routers where *openwrt* are installed for radio and link management.

For comparison, we employ two CSI-based multi-path estimators in our evaluations as baselines:

- **mD-Track** [53]: mD-Track is a state-of-the-art CSI-based multi-path estimator which extracts the ToF, AoA, AoD, and Doppler from CSI.
- **SpotFi** [28]: SpotFi uses the subspace-based 2D MUSIC algorithm to jointly estimate ToF and AoA from CSI.

Figure 6 shows the integrated result of the parameter estimation accuracy.

With the same RF configuration, the accuracy of BeamSense is close to SOTA CSI-based estimator. We use the CSI measurements from ath9k to obtain the baseline of CSI-based estimators, and use bidirectional CBR generated from CSI to evaluate the performance of BeamSense under the same RF configuration. As seen, although CBR compresses information from CSI, BeamSense can still achieve median errors in estimating the angular parameters (AoA/AoD) of the LOS path within 5°, angular parameters of the reflection path within 8°, and relative delay between two paths within 2ns. This result is close to that of mD-Track with complete CSI, where the corresponding median errors are 3.68°, 7.72°, and 1.61ns. Besides, both BeamSense and mD-Track are more accurate than SpotFi in all experiments.

The overall performance of BeamSense in 802.11ac/ax devices outperforms CSI-based estimators using 802.11n devices, due to a larger bandwidth used. Using captured CBR from 802.11ac/ax devices RTL8814 and MT76, BeamSense can be more accurate than mD-Track with ath9k, where the median estimation errors are reduced to 2.67°, 6.28° and 0.91ns, respectively. Note that Eq. 10 is a joint optimization of attenuation, angular, and delay parameters. Therefore, the estimation accuracy for each individual parameter can

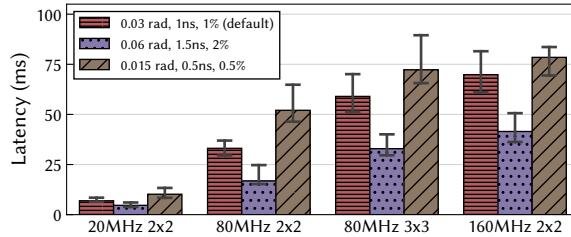


**Figure 8: Floorplan of indoor positioning experiments.** ▲ shows the locations of AP devices for establishing sensing links. ● shows the tested locations in device-based localization and passive tracking.

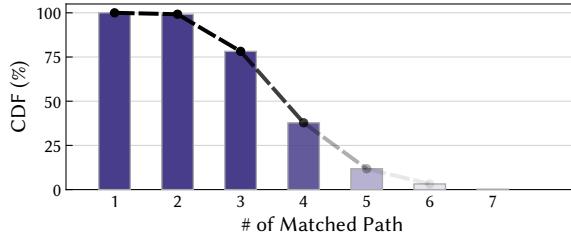
benefit from the larger bandwidth used in 802.11ac/ax devices. As one advantage provided by BeamSense is the compatibility with prevalent devices, this result demonstrates a potential benefit for existing sensing systems to deploy on these devices.

We next compare BeamSense with CBR-MUSIC, a sensing algorithm that estimates LOS AoD directly from CBR using MUSIC [22]. The MUSIC algorithm requires the knowledge of channel correlation matrix that cannot be derived from CBR. Therefore, CBR-MUSIC only approximates the central frequency correlation matrix by averaging all subcarriers' CBR. As shown in Fig. 11, this approximation yields large estimation errors, where the median AoD estimation errors of CBR-MUSIC are 4.17x and 4.86x higher than BeamSense for MT76 and RTL8814, respectively. This result demonstrates the significant advantage of BeamSense over directly estimating channel properties using the limited information of CBR.

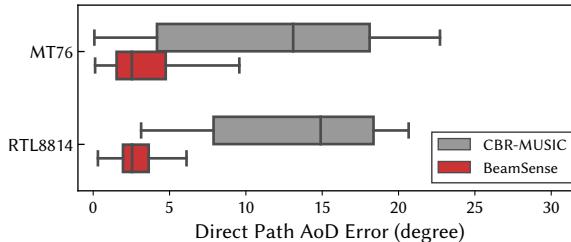
**Path matching.** In most cases, BeamSense and CSI-based estimator can obtain the same top-3 dominant paths. We study to what extent BeamSense can enable existing CSI-based algorithm work on CBR by counting the matched paths between the result of mD-Track using raw CSI and CSI generated by BeamSense. We regard two paths as *matched* if the differences of all parameters are less than 5%. As shown in Figure 10, in nearly all samples (99.2%), accurate LOS and reflection path parameters are obtained by BeamSense and mD-Track. In particular, in 78.2% of the total samples, BeamSense and mD-Track share the same top-3 dominant paths. The similarity



**Figure 9:** End-to-end latency of BeamSense under different RF and grid settings.



**Figure 10:** Distribution of matched path using mD-Track and BeamSense.



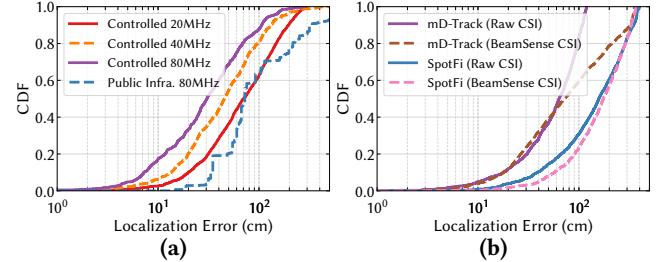
**Figure 11:** Direct path AoD estimation error of CBR-MUSIC and BeamSense.

drops distinctly after the fourth path. Because in many existing Wi-Fi sensing algorithms only the first several dominant components are considered [49], this result shows that BeamSense is capable to achieve a comparative performance in these tasks.

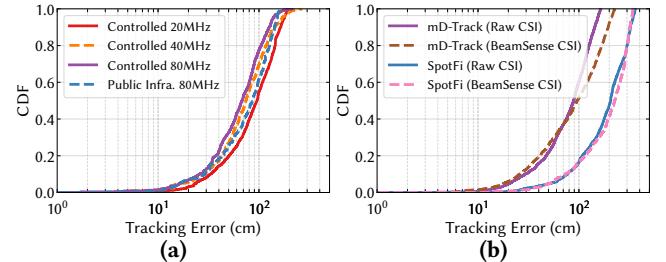
**Computational overhead.** Even with high-dimensional RF configurations, BeamSense could perform multi-path estimation within 80ms. The complexity of multi-path estimation scales with the number of subcarriers, the dimension of MIMO, and the grid space setting. Thus, we examine the end-to-end latency of BeamSense to a single multi-path profile from a pair of bidirectional CBR with different configurations. Figure 9 shows the integrated result. As expected, the time to perform multi-path estimation increases with the bandwidth, number of antennas, and grid space. Even so, the end-to-end latency of BeamSense with 234 subcarriers, 3×3 MIMO, and default grid setting is only 63ms, which will not impose much load on applications with stringent real-time requirements.

### 7.3 Device Localization

**Evaluation setup.** We demonstrate an application to localize connected devices with the multi-path estimations from BeamSense. During the experiment, each user at the test location holds a Wi-Fi device paired with the AP device. Based on the estimated LOS parameters and the actual location of the AP device, the location of



**Figure 12:** Device Localization Accuracy. (a) Performance of BeamSense in different testbed. (b) CSI-based methods with BeamSense CSI.



**Figure 13:** Passive Tracking Accuracy. (a) Performance of BeamSense in different testbeds. (b) CSI-based methods with BeamSense CSI.

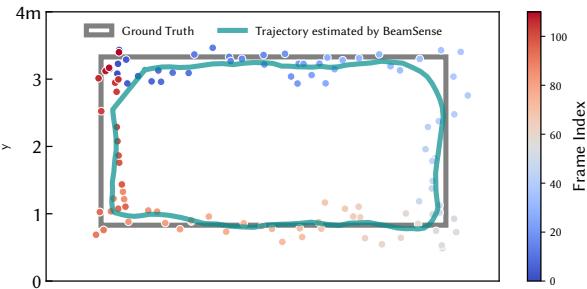
the user in the 2D plane is inferred. Figure 8 depicts the test locations and AP locations in controlled and public testbeds. We use the same CSI-based methods stated in §7.2 to estimate the multi-path parameters, and build the localization application according to [15].

**Overall performance.** BeamSense could achieve localization accuracy with 0.3~0.7 m median error in the controlled testbed and 0.72 m median error in the public testbed. Figure 12(a) shows the localization errors of BeamSense with different settings. In general, the precision increases with a larger bandwidth, where BeamSense is able to achieve a median error of 0.3 m, 0.45 m, and 0.7 m in the controlled testbed using 80MHz, 40MHz, and 20MHz bandwidth, respectively. However, as we cannot obtain the accurate antenna pattern of public devices, the median error with public infrastructure using 80MHz bandwidth increases by 140% (0.72 m).

**CSI-based methods could achieve a similar performance between using original CSI and BeamSense CSI.** The estimation of BeamSense inevitably contains inaccuracies. To study how it affects the downstream applications, we compare the localization error of CSI-based methods using original CSI and BeamSense CSI estimated from CBR on ath9k. Figure 12(b) shows the localization error with two different sources. As seen, the median localization errors of mD-Track using raw CSI and BeamSense CSI are 0.63 m and 0.72 m, where the difference is within 10 cm. SpotFi using raw CSI achieves a median error of 1.57 m, while the median error using BeamSense CSI is 1.75 m, where the difference is around 10%. Albeit BeamSense CSI diffuses the estimation errors, a similar performance is achieved.

### 7.4 Passive Tracking

**Evaluation setup.** We demonstrate an application to passively track the trajectory of the user near the sensing link. This application captures path profiles with similar Doppler characteristics and



**Figure 14: Example of passive tracking using BeamSense.**

estimates a trajectory using optimization methods introduced in [37]. We use the same apparatus stated in §7.3 and ask a user to walk near the sensing link following designated trajectories. An example trajectory is shown in Figure 14. A user walks following a rectangle, and dots show the BeamSense estimated user positions over time, based on which the trajectory of the user is estimated.

**Overall performance.** *BeamSense archives a median error of 0.67 ~0.95 m across different settings.* As shown in Figure 13(a), in both controlled and public testbeds, BeamSense achieve a stable performance with median errors ranging from 0.67 m to 0.95 m. With the bandwidth increase from 20MHz to 40MHz, we can observe a significant improvement (21.97%) in tracking accuracy, but this improvement from 40MHz to 80MHz is decreased (8.7%). While the higher bandwidth enables more accurate delay estimates, the tracking errors are dominated by the angle estimation error after increasing to a certain bandwidth.

We conduct the same experiment as in §7.3 to evaluate the CSI-based methods with BeamSense CSI, where Figure 13(b) shows a result similar to previous discussions.

## 7.5 Sign Language Recognition

**Evaluation setup.** We demonstrate a sign language recognition application using CSI estimated by BeamSense. The experiments involve 3 users, and we let each user stand nearby the sensing link to perform 20 different sign words. Each sign word is repeated 10 times and elapses 1~2 seconds. We use a pair of RTL8814 interface cards to report CBR during this period. In total, we collect 600 segmented bidirectional CBR series for testing.

We build a simple 9-layer Convolution Neural Network (CNN) for classification. To examine the cross-domain performance, we consider three different types of training/testing datasets. The first type of dataset uses samples from the same user for both training and testing. The second type follows a cross-subject manner, where samples from two of the three users are used for training, and samples from the other user are used for testing. The third type mixes all data samples of three users. Across all configurations, the ratio between training and testing samples is 3 : 1.

**Overall performance.** Figure 16 shows the integrated recognition accuracy of the CNN classifier with BeamSense estimated channel states. For comparison, we build another CNN classifier with the same network architecture while using raw CBR frames only.

*CNN classifier with BeamSense estimated channel state achieves 92.5~97.14% accuracy with in-subject samples.* As shown, for datasets comprising samples from each individual user, the CNN classifier with BeamSense estimated CSI can achieve 92.5%, 95%, and 97.14%

accuracy, respectively, compared with 85%, 72.5%, and 85.71% for CNN with raw CBR. Compared to directly using CBR frames, BeamSense could improve the accuracy by 17.51% on average, which demonstrates that the primitive features extracted by BeamSense are more effectively depicting human activities.

*BeamSense achieves around 70% recognition accuracy in cross-subject samples, which vastly improves the cross-domain capability of sensing with CBR.* For the dataset comprising cross-subject samples, the CNN classifier with raw CBR only achieves 4.49% accuracy, which is similar to a result of random guessing. This is because the same gesture performed by different people typically can produce different channel decomposition results. As a result, the ML model rarely learns cross-domain knowledge from CBR frames. On the other hand, the CNN classifier underpinned by BeamSense can achieve an accuracy of 70%. This is because BeamSense can accurately reconstruct the multi-path channel from CBR, making it more robust across different deployment environments and use scenarios. It is also corroborated in the mixed dataset, where BeamSense (92.17%) improves the accuracy by 45.1% as of using raw CBR (63.48%).

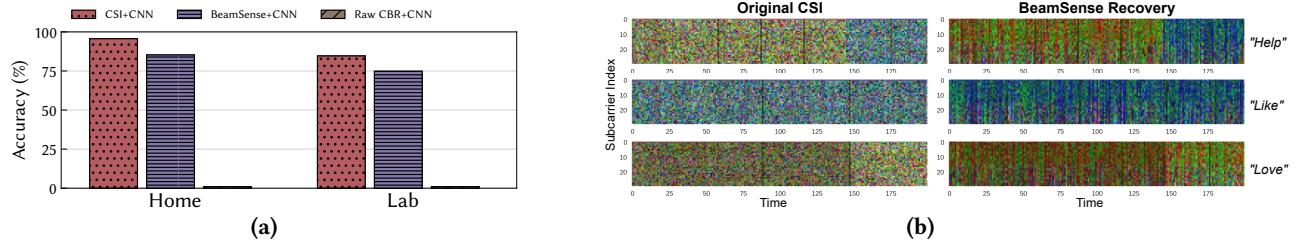
**Performance on *SignFi* dataset.** We also examine the classification performance on the *SignFi* dataset [33]. The dataset consists of 15,780 samples of CSI series for recognizing 276 sign languages, captured in 2 different environments. We generate bidirectional CBR from each CSI frame and use the identical CNN classifier to evaluate the accuracy with raw CSI, raw CBR, and BeamSense estimated CSI. Figure 15(a) shows the integrated result.

*Raw CBR is ill-suited in this task with a large number of classes, while BeamSense can still achieve an average accuracy of 80.09%, which only drops around 10% compared to using original CSI.* As seen, for both two environments, the performance of using raw CBR drops severely due to the increased number of output classes. On the other hand, using the information extracted by BeamSense can achieve 85.27% and 74.91%, which only drops around 10% from using the original CSI (95.64% and 84.64%). This result demonstrates the scheme of BeamSense could enable sensing with CBR even for tasks with increased difficulty.

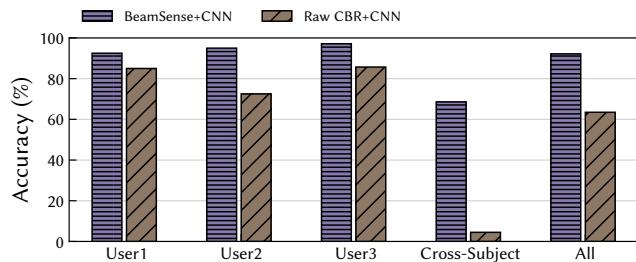
Figure 15(b) shows an example of BeamSense recovered CSI series representing three sign words, ‘Help’, ‘Like’, and ‘Love’. Each (time, subcarrier) pixel in the RF image is rendered based on the normalized RX gains of three antennas on the AP device (*i.e.*, mapping to three RGB channels). Similar to image recognition methods, the CNN classifier learns different gestures by recognizing the low-level features of the RF image, *e.g.*, blockage patterns (the occasional dark lines in the figure), colors, and duration of each segment. As shown in Figure 15(b), the RF images generated by BeamSense exhibit similar patterns to that of the original CSI. Specifically, the cosine similarity between the datasets of BeamSense-CSI and the original CSI is 0.832. This demonstrates why BeamSense-underpinned CNN is able to achieve a similar performance to that of CNN operated on the original CSI.

## 8 Discussion

**Number of TX/RX antennas.** In certain Wi-Fi router deployments, the device may use more antennas in reception (*e.g.*, 4) while using fewer antennas in transmission (*e.g.*, 3) for the sake of intensifying the uplink performance. Under this circumstance, the



**Figure 15:** Sign language recognition with CBR generated from *SignFi* dataset. (a) Recognition accuracy. (b) Comparison between the original CSI and the CSI recovered by BeamSense, where pixel colors represent the normalized RX gains of 3 antennas on the access point. The dark lines are caused by occlusions.



**Figure 16:** Sign language recognition in our testbed.

intrinsic fingerprint of downlink and uplink channels will not be identical and thus the fingerprint test matrix cannot be constructed. Therefore, as a requirement of BeamSense, each device should use the number of antennas in both transmission and reception, which is configurable on most Wi-Fi devices via the mac80211 interface.

**Handling absence of beamformer capability.** BeamSense leverages bidirectional CBR to estimate the multi-path channel. Although the STA beamformee capability is widely present, the STA beamformer capability is rare. In our city-scale measurement, only 1.0% STAs report beamformer capability. To collect bidirectional CBR in this case, a simple workaround is to inject NDPA and NDP frames via packet injection and emulation [14] from STA to trigger uplink channel sounding.

**Privacy concerns.** Sniffing on-the-air CBR frames may raise privacy concerns and even disclose the critical physical features of the channel. To resolve this issue, an effective solution is to enforce WPA3 [7] in the connection, which will encrypt all management frames including the CBR between two WiFi devices. In such cases, only authorized devices that hold the encryption key can decrypt the CBR from wireless traffic. However, this would not affect the applicability of our system, because legitimate users usually associate with the target AP.

## 9 Conclusion

In this paper, we propose the first generalized Wi-Fi sensing paradigm based on compressed beamforming reports (CBR). We devise a novel framework that is computationally efficient to map CBR to a multi-path profile. We implement BeamSense on several prevalent models of Wi-Fi devices and evaluate the performance of BeamSense with microbenchmarks and three representative Wi-Fi sensing applications. Our result shows BeamSense can achieve high sensing accuracy and superior generalizability.

## ACKNOWLEDGEMENT

We sincerely thank our shepherd and the anonymous reviewers for their valuable feedbacks and constructive suggestions. This work is supported in part by Research Grants Council (RGC) of Hong Kong under General Research Fund No. 14211121, No. 14209619, and No. 11204722.

## References

- [1] EDUP-1622 802.11ac Dual Band WIFI USB 3.0 Adapter. <https://www.szedup.com/product-item/802-11ac-dual-band-wifi-usb-3-0-adapter>.
- [2] Linux Wireless: ath9k Products. <https://wireless.wiki.kernel.org/en/users/drivers/ath9k/products>.
- [3] Netgear WNDR4300 v1. <https://openwrt.org/toh/netgear/wndr4300>.
- [4] OpenWrt Wiki - Supported devices. <https://openwrt.org/supported-devices>.
- [5] Ruijie X32 Pro. <https://www.ruijenetworks.com/products/REYEE-MeshWi-Fi/RG-EW1200-Series/RG-EW3200GX-PRO>.
- [6] Tenda AC2100 Dual Band Gigabit WiFi Router. <https://www.tendacn.com/product/AC23.html>.
- [7] WPA3 Specification. <https://www.wi-fi.org/file/wpa3-specification>.
- [8] Xiaomi Mi Router CR6608. [https://openwrt.org/toh/hwdata/xiaomi/xiaomi\\_mi\\_router\\_cr6608](https://openwrt.org/toh/hwdata/xiaomi/xiaomi_mi_router_cr6608).
- [9] IEEE Standard for Information technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. *IEEE Std 802.11ac(TM)-2013*, pages 1–425, Dec 2013.
- [10] Fadel Adib and Dina Katabi. See through Walls with WiFi! In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, page 75–86, New York, NY, USA, 2013. Association for Computing Machinery.
- [11] P. Bahl and V.N. Padmanabhan. RADAR: An In-building RF-based User Location and Tracking System. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 2, pages 775–784 vol.2, 2000.
- [12] Arjun Bakshi, Yifan Mao, Kannan Srinivasan, and Srinivasan Parthasarathy. Fast and Efficient Cross Band Channel Prediction Using Machine Learning. In *The 25th Annual International Conference on Mobile Computing and Networking, MobiCom '19*, New York, NY, USA, 2019. Association for Computing Machinery.
- [13] D. I. Cartwright and M. J. Field. A Refinement of the Arithmetic Mean-Geometric Mean Inequality. *Proceedings of the American Mathematical Society*, 71(1):36–38, 1978.
- [14] Ruirong Chen and Wei Gao. TransFi: Emulating Custom Wireless Physical Layer from Commodity WiFi. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services, MobiSys '22*, page 357–370, New York, NY, USA, 2022. Association for Computing Machinery.
- [15] Zhe Chen, Guorong Zhu, Sulei Wang, Yuedong Xu, Jie Xiong, Jin Zhao, Jun Luo, and Xin Wang.  $M^3$ : Multipath Assisted Wi-Fi Localization with a Single Access Point. *IEEE Transactions on Mobile Computing*, 20(2):588–602, 2021.
- [16] Rui Du, Hailiang Xie, Mengshi Hu, Yan Xin, Stephen McCann, Michael Montemurro, Tony Xiao Han, Jie Xu, et al. An Overview on IEEE 802.11 bf: WLAN Sensing. *ArXiv preprint arXiv:2207.04859*, 2022.
- [17] DeviWiki (ex WikiDevi). List of Wi-Fi Device IDs in Linux, Oct 2016. [https://deviwiki.com/wiki/List\\_of\\_Wi-Fi\\_Device\\_IDs\\_in\\_Linux](https://deviwiki.com/wiki/List_of_Wi-Fi_Device_IDs_in_Linux).
- [18] Chuhan Gao, Yilong Li, and Xinyu Zhang. LiveTag: Sensing Human-Object Interaction through Passive Chipless WiFi Tags. In *15th USENIX Symposium on*

- Networked Systems Design and Implementation*, NSDI '18, pages 533–546, Renton, WA, April 2018. USENIX Association.
- [19] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*, WiTECH '19, page 21–28, New York, NY, USA, 2019. Association for Computing Machinery.
- [20] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool Release: Gathering 802.11n Traces with Channel State Information. *SIGCOMM Comput. Commun. Rev.*, 41(1):53, jan 2011.
- [21] Yi Hong. *Delay-Doppler Communications: Principles and applications*. Elsevier Science and Technology, 2022.
- [22] Sohei Itahara, Sota Kondo, Kota Yamashita, Takayuki Nishio, Koji Yamamoto, and Yusuke Koda. Beamforming Feedback-Based Model-Driven Angle of Departure Estimation Toward Legacy Support in WiFi Sensing: An Experimental Study. *IEEE Access*, 10:59737–59747, 2022.
- [23] Woojae Jeong, Jinwhan Jung, Yuanda Wang, Shuai Wang, Seokwon Yang, Qiben Yan, Yung Yi, and Song Min Kim. SDR Receiver Using Commodity WiFi via Physical-Layer Signal Reconstruction. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.
- [24] Takamichi Kanda, Takashi Sato, Hiromitsu Awano, Sota Kondo, and Koji Yamamoto. Respiratory Rate Estimation Based on WiFi Frame Capture. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 881–884. IEEE, 2022.
- [25] Sorachi Kato, Takuma Matsukawa, Tomoki Murakami, Takuji Fujihashi, Takashi Watanabe, and Shunsuke Saruwatari. Fundamental Investigation of Wi-Fi Beamforming Report Properties on Wireless Sensing. In *2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 341–344. IEEE, 2022.
- [26] Sorachi Kato, Tomoki Murakami, Takuji Fujihashi, Takashi Watanabe, and Shunsuke Saruwatari. CBR-ACE: Counting Human Exercise Using Wi-Fi Beamforming Reports. *Journal of Information Processing*, 30:66–74, 2022.
- [27] Sota Kondo, Sohei Itahara, Kota Yamashita, Koji Yamamoto, Yusuke Koda, Takayuki Nishio, and Akihito Taya. Bi-directional Beamforming Feedback-based Firmware-agnostic WiFi sensing: An Empirical Study. *IEEE Access*, 10:36924–36934, 2022.
- [28] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadwaj, and Sachin Katti. SpotFi: Decimeter Level Localization Using WiFi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 269–282, New York, NY, USA, 2015. Association for Computing Machinery.
- [29] Yuanjie Li, Qianru Li, Zhehui Zhang, Ghufran Baig, Lili Qiu, and Songwu Lu. Beyond 5G: Reliable Extreme Mobility Management. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '20, page 344–358, New York, NY, USA, 2020. Association for Computing Machinery.
- [30] Zhuqi Li, Yaxiong Xie, Longfei Shangguan, Rotman Ivan Zelaya, Jeremy Gummesson, Wenjun Hu, and Kyle Jamieson. Towards Programming the Radio Environment with Large Arrays of Inexpensive Antennas. In *16th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '19, pages 285–300, Boston, MA, February 2019. USENIX Association.
- [31] Scapy Library. <https://scapy.net/>.
- [32] Yongsen Ma, Gang Zhou, and Shuangquan Wang. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.*, 52(3), jun 2019.
- [33] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. SignFi: Sign Language Recognition Using WiFi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1), mar 2018.
- [34] Francesca Meneghelli, Michele Rossi, and Francesco Restuccia. DeepCSI: Rethinking Wi-Fi Radio Fingerprinting Through MU-MIMO CSI Feedback Deep Learning. In *2022 IEEE 42nd International Conference on Distributed Computing Systems*, ICDCS '22. IEEE, jul 2022.
- [35] Kazuki Noguchi, Osamu Muta, Tomoki Murakami, and Shinya Otsuki. A CSI-based Object Detection Scheme using Interleaved Subcarrier Selection in Wireless LAN Systems with Distributed Antennas. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 01–05, Sep. 2021.
- [36] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. Widar: Decimeter-level Passive Tracking Via Velocity Monitoring with Commodity WiFi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '17, pages 1–10, 2017.
- [37] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. Widar2.0: Passive Human Tracking with A Single Wi-Fi Link. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys'18, pages 350–361, 2018.
- [38] R. Schmidt. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986.
- [39] Ken Shoemake. Animating Rotation with Quaternion Curves. *SIGGRAPH Comput. Graph.*, 19(3):245–254, jul 1985.
- [40] Sanjib Sur, Xinyu Zhang, Parmesh Ramanathan, and Ranveer Chandra. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *13th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '16, pages 193–206, 2016.
- [41] Navod Suraweera, Alycia Winter, Julian Sorensen, Mark Johnson, Shenghong Li, Iain B. Collings, Stephen V. Hanly, Wei Ni, and Mark Hedley. Stand-off Detection of Human Presence and Movement Using IEEE 802.11ac Beamforming Reports. In *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–7, 2019.
- [42] Navod Suraweera, Alycia Winter, Julian Sorensen, Shenghong Li, Mark Johnson, Iain B. Collings, Stephen V. Hanly, Wei Ni, and Mark Hedley. Passive Through-Wall Counting of People Walking Using WiFi Beamforming Reports. *IEEE Systems Journal*, 15(4):5476–5482, 2021.
- [43] Sheng Tan, Yili Ren, Jie Yang, and Yingying Chen. Commodity WiFi Sensing in Ten Years: Status, Challenges, and Opportunities. *IEEE Internet of Things Journal*, 9(18):17832–17843, 2022.
- [44] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, USA, 2005.
- [45] Deepak Vasishtha, Swaran Kumar, and Dina Katabi. Decimeter-Level Localization with a Single WiFi Access Point. In *13th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '16, pages 165–178, Santa Clara, CA, March 2016. USENIX Association.
- [46] Deepak Vasishtha, Swaran Kumar, Hariharan Rahul, and Dina Katabi. Eliminating Channel Feedback in Next-Generation Cellular Networks. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, page 398–411, New York, NY, USA, 2016. Association for Computing Machinery.
- [47] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. LiFS: Low Human-effort, Device-free Localization with Fine-grained Subcarrier Information. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, MobiCom '16, pages 243–256, 2016.
- [48] Song Wang, Jingqi Huang, and Xinyu Zhang. Demystifying Millimeter-Wave V2X: Towards Robust and Efficient Directional Connectivity under High Mobility. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New York, NY, USA, 2020. Association for Computing Machinery.
- [49] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, page 65–76, New York, NY, USA, 2015. Association for Computing Machinery.
- [50] Stephen J. Wright. Coordinate Descent Algorithms. *Math. Program.*, 151(1):3–34, jun 2015.
- [51] Wei Xi, Chen Qian, Jinsong Han, Kun Zhao, Sheng Zhong, Xiang-Yang Li, and Jizhong Zhao. Instant and Robust Authentication and Key Agreement among Mobile Devices. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 616–627, New York, NY, USA, 2016. Association for Computing Machinery.
- [52] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise Power Delay Profiling with Commodity WiFi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, page 53–64, New York, NY, USA, 2015. ACM.
- [53] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. MD-Track: Leveraging Multi-Dimensionality for Passive Indoor Wi-Fi Tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.
- [54] Zheng Yang, Zimu Zhou, and Yunhao Liu. From RSSI to CSI: Indoor Localization via Channel Response. *ACM Comput. Surv.*, 46(2), dec 2013.
- [55] Moustafa Youssef and Ashok Agrawala. The Horus WLAN Location Determination System. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, page 205–218, New York, NY, USA, 2005. Association for Computing Machinery.
- [56] Youwei Zeng, Dan Wu, Jie Xiong, Enze Yi, Ruiyang Gao, and Daqing Zhang. FarSense: Pushing the Range Limit of WiFi-Based Respiration Sensing with CSI Ratio of Two Antennas. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(3), sep 2019.
- [57] Yunze Zeng, Parth H. Pathak, and Prasant Mohapatra. WiWho: WiFi-Based Person Identification in Smart Spaces. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '16, pages 1–12, 2016.
- [58] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Zero-Effort Cross-Domain Gesture Recognition with WiFi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, page 313–325, New York, NY, USA, 2019. Association for Computing Machinery.