mm-FLEX: An Open Platform for Millimeter-Wave Mobile Full-Bandwidth Experimentation

Jesus Omar Lacruz
jesusomar.lacruz@imdea.org
IMDEA Networks Institute
Madrid, Spain

Dolores Garcia
dolores.garcia@imdea.org
IMDEA Networks Institute
Universidad Carlos III de Madrid
Madrid, Spain

Pablo Jimenez
pablo.jimenez@imdea.org
IMDEA Networks Institute
Universidad Carlos III de Madrid
Madrid, Spain

Joan Palacios
joan.palacios@imdea.org
IMDEA Networks Institute
Universidad Carlos III de Madrid
Madrid, Spain

Joerg Widmer
joerg.widmer@imdea.org
IMDEA Networks Institute
Madrid, Spain

ABSTRACT

Millimeter-Wave (mm-wave) technology is increasingly being considered for mobile devices and use cases such as vehicular communication. This requires suitable experimentation platforms to support systems-oriented research to tackle the multitude of problems and challenges of mm-wave communications in such environments. To this end, we introduce mm-FLEX, a flexible and modular open platform with real-time signal processing capabilities that supports a bandwidth of 2 GHz and is compatible with mm-wave standard requirements. mm-FLEX integrates an FPGA-based baseband processor with full-duplex capabilities together with mm-wave RF front-ends and phased antenna arrays that are fully configurable from the processor in real-time. To demonstrate the capabilities of mm-FLEX, we implement a scalable, ultra-fast beam alignment mechanism for IEEE 802.11ad systems. It is based on compressive estimation of the signal's angle-of-arrival by means of switching through multiple receive beam patterns on a nano-second timescale while receiving a packet preamble. Our implementation is open source and is made publicly available to the research community.

CCS CONCEPTS

• **Networks** → **Network experimentation**; **Mobile networks**; • **Computer systems organization** → **Real-time system architecture**; • **Hardware** → **Digital signal processing**.

KEYWORDS

Millimeter Wave, Testbed, FPGA, Phased Antenna Array, Beam Training

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.
MobiSys '20, June 15–19, 2020, Toronto, ON, Canada
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7954-0/20/06…$15.00
https://doi.org/10.1145/3386901.3389034

ACM Reference Format:
Jesus Omar Lacruz, Dolores Garcia, Pablo Jimenez, Joan Palacios, and Joerg Widmer. 2020. mm-FLEX: An Open Platform for Millimeter-Wave Mobile Full-Bandwidth Experimentation. In *The 18th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '20), June 15–19, 2020, Toronto, ON, Canada.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3386901.3389034

1 INTRODUCTION

Millimeter-Wave (mm-wave) network technology such as the IEEE 802.11ad [33] and the upcoming IEEE 802.11ay [34] standards for 60 GHz Wireless Local Area Networks (WLANs) can provide vastly higher data rates than networks operating below 6 GHz, due to the large amount of bandwidth available at mm-wave frequencies. Mm-wave devices typically implement directional communication using phased arrays, so that the beam-forming gains of the antenna compensate for the higher path loss due to the small wavelength [22]. This not only ensures a good link range but also reduces interference and thus improves spatial reuse. Operating networks at mm-wave frequencies introduces new problems not only due to the RF front-end components but also due to the need for high gain directional communication and the associated beam training or channel estimation. Especially for mobile scenarios, further practical research is needed. For instance, fast device tracking is essential for continuous antenna beam alignment, but current beam training mechanisms introduce a high overhead and latency when used in highly mobile environments. At the same time, there is a lack of suitable experimentation platforms that allow for real-time packet processing and antenna reconfiguration, and that are flexible enough to support modifications of the physical and MAC layers. To address the challenges of mobile mm-wave communication networks, it is crucial to provide the research community with such an experimentation platform, that ideally should have the following features: i) giga-sampling rate capability, to be able to accurately study the characteristics of the wide-band channels and be compatible to current standards, ii) access to physical layer data, which brings invaluable information to researchers to measure the channel and develop new signal processing algorithms, iii) simplicity, flexibility and configurability, such that researchers can easily modify the behavior of the system, iv) fast antenna reconfiguration

to support high-mobility scenarios, and v) affordability, to enable also smaller research groups to carry out mobile systems research. The latter is a problem specific to mm-wave systems and their large bandwidth, since the cost of the A/D converters and the FPGA are directly related to the maximum sampling frequency and the number of logic elements/speed grade, respectively.

Different solutions in the literature aim to address the need for experimentation platforms. The MiRa platform [2] uses Universal Software Radio Peripheral (USRP) devices as baseband equipment attached to a custom 28 GHz RF front-end. It only supports narrow-band channels due to the USRP's bandwidth limitations. The X60 testbed [23] is based on National Instruments (NI) hardware and a phased array antenna kit. While it meets the bandwidth requirements of the IEEE 802.11ad standard, it is expensive and the high complexity of the design makes low-level modifications more difficult. OpenMili [35] aims to increase performance compared to [2] at a more manageable cost compared to [23]. It does provide better performance than [2] but still does not meet the bandwidth requirements of mm-wave standards. Our system is complementary to [36] which provides a MIMO-capable RF front-end, whereas our focus is on a full-bandwidth baseband design. In addition, the solutions listed above lack support for high-mobility scenarios, which require fast control over the antenna array to enable real-time beam steering with very low antenna reconfiguration latency.

In this paper we propose a *M*illimeter Wave *M*obile *F*ul*L*-Bandwidth *EX*perimentation Platform (mm-FLEX). It is a flexible, modular and highly configurable testbed that exceeds the requirements of current mm-wave standards, supporting more than 2 GHz of bandwidth. mm-FLEX is an open source research platform that offers a unique set of features. At approximately twice the cost of a high-end narrow-band USRP (an order of magnitude less than [23]), it offers more than ten times the performance and does not sacrifice standard compliance as [2, 35]. Furthermore, we add important features such as fast real-time antenna steering that are a key enablers for research targeting high mobility scenarios.

The testbed is composed of a powerful baseband system including a high performance FPGA and processor boards with the necessary components for real-time signal processing, as well as a 60 GHz RF front-end using a 16+16 element phased array antenna with high resolution phase shifters that can be fully controlled from the system. We design mm-FLEX to be modular and flexible with the necessary IP blocks that facilitate real-time experimentation with mobile mm-wave systems. Furthermore, the designed blocks have a flexible AXI interface that allows different Super Sample Rate (SSR) configurations such that they can even be used on different FPGA platforms, such as the narrow-band X310 USRP system. We implement the signal processing blocks for real-time capture and frame processing directly on the FPGA[1], together with the fast beam sweeping of the phased-array antenna at nano-second time scale. Our testbed also supports a hardware-in-the-loop mode where part of the functionality is software modeled while the time-critical ones are hardware-implemented, facilitating the development and validation of signal processing applications for the mm-wave community.

[1]For simplicity, in this work only the preamble is processed on the FPGA in real-time, while frame decoding is done offline. However, the FPGA is large enough for a standard-compliant real-time IEEE 802.11ad transceiver and we intend to implement this in the future.

While our experimentation platform is currently designed for IEEE 802.11ad, it is flexible enough to be used with 28GHz equipment [27] for 5G New Radio (NR) and beyond research.

To showcase the capabilities of mm-FLEX, we use it to address the important problem of fast beam alignment in dense, mobile mm-wave scenarios. Conventional IEEE 802.11ad [33] beam training is done by transmitting frames using each of the available antenna configurations, while the target device listens with a quasi-omnidirectional beam pattern. This avoids the complexity of training both ends of the link simultaneously. The target device then provides feedback which sector resulted in the highest Signal-to-Noise Ratio (SNR) [14]. Such brute force or simple hierarchical beam training (in case beam refinement is used) works well for relatively static deployments of moderate density. However, the overhead may become prohibitive when frequent beam training is required in more dynamic environments with device mobility or link interruptions through blocking obstacles.

We propose a design that *eliminates dedicated station beam training* to allow the network to handle very high network dynamics and client densities. Our mechanism works with unmodified standard-compliant IEEE 802.11ad Access Points (APs). While an AP sends multiple frames for the standard beam training, the stations perform quick signal strength measurements with different receive beam patterns during preamble reception of these frames after packet detection. We then use a fast sparse estimation algorithm to determine the Angle of Arrival (AoA) and select the best beam pattern to be used at the station. Our mechanism only requires minor changes in the firmware of the stations, since the fast beam-pattern sweeping capability is already included in 802.11ad standard compliant devices for the Beam Refinement Phase (BRP). We perform an extensive measurement campaign and show that our mechanism can switch through 10 receive beam patterns during the preamble without compromising detection accuracy for typical SNR values, whereas only 5 beam patterns are required to accurately estimate the AoA with an error of around 1°.

Overall, our paper provides the following contributions:

- We design mm-FLEX, a flexible and configurable testbed that is able to perform real-time signal processing at multi-GHz bandwidths and is compatible with current mm-wave standards. At the same time, the hardware on which the system is based is affordable for such a high performance system.
- We implement the baseband system on a powerful FPGA and processor board with dual AXI-stream data paths for separate transmitter and receiver chains. This brings modularity to the design and enables full-duplex applications. Our hardware processing blocks to detect and process the preamble of standard-compliant IEEE 802.11ad frames operates with the full bandwidth of 1.76 GHz. The code and specification of our testbed are publicly available [6].
- We augment the testbed functionality to provide fast real-time antenna array reconfigurability, implementing a hardware block to directly switch beam patterns on a nano-second time scale, using a configurable input trigger. This is a crucial feature not only to support standard-compliant IEEE 802.11ad beam training, but more importantly to enable symbol-level beam reconfiguration for research targeting high-mobility scenarios.
- We demonstrate the capabilities of the testbed platform by implementing a mechanism for passive station beam alignment for dense,

mobile millimeter-wave networks, that significantly outperforms current standards and prior work.

2 PLATFORM ARCHITECTURE

In this section, we describe the top-level hardware architecture of mm-FLEX and the design choices taken that ensure flexibility and modularity of the system.

2.1 Hardware components

The proposed testbed system is composed of two main components, i) a powerful baseband processor that can generate and capture baseband IQ samples at Giga-sampling rates from a ii) 60GHz RF front-end with phased array antennas with analog beam-forming capabilities, which performs direct up/down conversion from/to the baseband samples. This design limits the number of hardware components required to implement the testbed. A simple block diagram is shown in Fig. 1. While we describe the testbed development with 60GHz transceivers, mm-FLEX can support RF front-ends for other mm-wave frequencies [27].

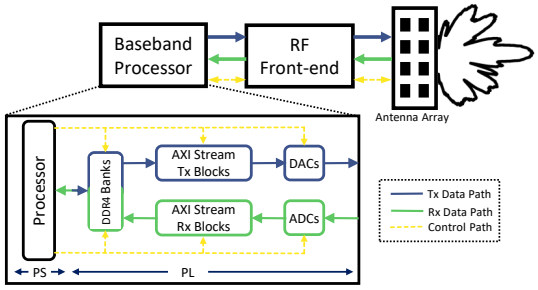


Figure 1: mm-FLEX top-level architecture showing Tx and Rx data paths and control path

The *baseband processor* consists of a single chassis including an i) AMC599 board, integrating a Xilinx Kintex Ultrascale FPGA, high-speed AD/DA converters and DDR4 memory banks, among other features [31] and ii) an AMC726 board with an Intel Core i7 processor [32]. The AMC599 board provides the Programmable Logic (PL) to implement the hardware signal processing blocks for processing the IQ samples that are sent to and captured from the converters and stored in the on-board DDR memory. The AMC726 board provides the Processing System (PS), serving as a management and configuration processor for the FPGA, AD/DA converters and RF front-end. The PL and PS subsystems are connected to each other through a PCIe interface in the chassis' back plane. The chassis allows to connect an additional AMC599 board that can be managed from the same PS via PCIe, which allows to extend mm-FLEX to support mm-wave MIMO communications, as used for example in the latest standards [34]. The *RF front-end* is composed of 60 GHz up/down converters with phased antenna arrays. Finally, an external dual channel clock board is used to generate *independent* sampling clock signals for the AD/DA converters (i.e., we do not require that the communicating transceivers are clock synchronized). A photo of the system with labels for the most relevant hardware components is shown in Fig. 2a. System specifications exceed by far the bandwidth and processing requirements for an IEEE 802.11ad compliant transceiver and offer enough flexibility to be used for more powerful future mm-wave communication standards.

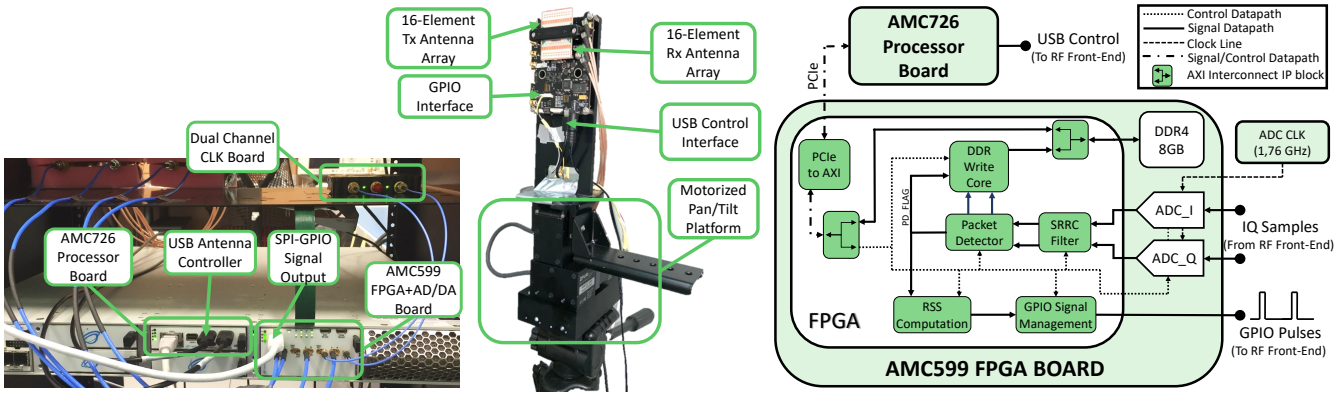
2.2 General data/control architecture

2.2.1 Data path. We develop mm-FLEX to have full-duplex capabilities and therefore design independent transmitter and receiver data paths, as shown with solid lines in Fig. 1). They move the IQ samples through the different signal processing blocks from the source, in the baseband processor, to the transmitter phased array antenna and vice versa on the receiver side. Each data path has an independent clocking structure, DDR memory and separated blocks to transmit and receive IQ samples from/to the converters, without sacrificing bandwidth or real-time signal processing capabilities.

To ensure modularity of the system, facilitate block interconnection and avoid inter-block synchronization issues, we adopt the well-known AXI-stream for both data paths (Fig. 1) to move the IQ samples through the signal processing blocks. Thanks to this interface, high-level tools like Xilinx Vivado HLS and System Generator can be used to develop custom signal processing blocks, providing a reasonably high level of abstraction to extend the design. With Giga-Samples per Second (GSPS) sampling frequencies of the AD/DA converters and a PL clock frequency on the order of hundreds of MHz, multiple samples must be processed in parallel during every clock cycle in each channel of the data paths. The number of parallel samples is defined by the SSR parameter, the ratio between the sampling clock frequency and the PL clock frequency. Here, we adopt an SSR of 16 parallel samples for full bandwidth operation. This parameter can be configured depending on the specific design requirements, in favor of area reduction or timing constraints relaxation. Although mm-FLEX can be used with a wide range of sampling clock frequencies to meet specific standard requirements, in the remainder of this paper the sampling frequency for the DACs/ADCs is selected to be 3.52 GSPS (1.76 Gsymbols/s * 2 samples/symbol, which gives a clock frequency of 220MHz for both data paths, considering a SSR of 16). This ensures that the system meets the symbol rate requirements of the IEEE 802.11ad standard of 1.76 Gsymbols/s, considering two samples per received symbol.

Communication between the PL and the PS is done by means of a PCI-to-AXI IP block that enables fast large data transfers from/to the on-board DDR memory, as required due to the Giga sampling rates. Access to the DDR memory from the PS and PL is done by means of custom-designed DDR write and DDR read hardware blocks. These efficient write/read operations to/from the on-board memory match the SSR adopted for the AXI-stream data-width in the design. Specifically, the DDR write core is user configurable from the PS to operate in two different modes: i) *simple writing*, where the user configures the amount of IQ samples to be stored in the DDR memory in a single capture (up to 250 million IQ samples) and ii) *burst saving*, where the user configures the number of chunks of data (and their length) that should be captured and saved for further processing. In this case, the block requires a hardware trigger input that indicates when to start saving each chunk.

2.2.2 Control path. System control and management is done entirely from the PS (AMC726 processor) through the PCIe interface. This allows not only to move large amounts of data to/from the



(a) Baseband processor (left) and 60 GHz RF front-end (right) **(b) Baseband receiver block diagram**

Figure 2: mm-FLEX main components

DDR memory but also provides control and management of the data converters, IP blocks and RF front-end. To this end we include an AXI protocol conversion in the PL, which passes from an AXI-stream to an AXI-lite interface, allowing to send and receive commands to the configuration registers of each of the modules, as shown in Fig. 1 (dotted lines). The PS controls the RF front-end by means of a USB interface and Python functions (supplied by the manufacturer), that allow to configure each kit as transmitter or receiver (at runtime), transfer the antenna codebook, set the Antenna Wave Vector (AWV), power amplifier gains, and carrier frequency, among other features.

From the user point of view, a C/C++ application is used to configure and operate the system through an SSH connection. Simple commands are used to run functions that entirely configure the baseband processor, send and receive IQ samples to/from DDR memory, configure IP blocks, etc. On system start-up, the user just runs an "init" function to configure the PLLs for the converters, to set up the transmitter and receiver data paths, and configure the peripherals of the system. Besides, the user can configure at any moment the IP blocks included in each one of the data paths by means of commands that write directly to the IP block registers using the AXI-lite bus.

2.3 Baseband processing blocks

Considering the simple block diagram of Fig. 1 and the system architecture described above, mm-FLEX is able to transfer complex IQ samples from the ADCs directly to one DDR memory at 6.4 GSPS, without the need for any custom AXI-stream block. At the same time, in the transmitter chain, the other DDR bank can stream IQ samples directly to the DACs. With this system structure, mm-FLEX can be used in a hardware-in-the-loop system where the baseband processor is used to transmit and capture IQ samples while the entire decoding is done in software, as is often done with conventional USRP devices. Although useful and simple, this operation mode does not fully exploit the capabilities of the powerful baseband processor of mm-FLEX. We develop the system to allow users to perform step-by-step translation of signal processing functions with heavy computational load to be hardware implemented, while the entire system (hardware-software) remains fully functional. This helps not only to accelerate the processing of the IQ samples but also allows hardware designers to validate the implementation of signal processing blocks in real-time environments, comparing their behavior with software floating point models.

Next we include the designed hardware blocks that facilitate experimentation with IEEE 802.11ad systems, translating part of the signal processing tasks to hardware implementation, while the rest of the decoder processing functions remain software-based. This allows potential users to translate more functions to the PL side using the provided system as a reference design to further upgrade and extend functionality. The designed blocks are flexible enough to be used with different baseband processors with different sampling rate requirements, since the SSR can be changed by means of a single parameter in the block interface.

• **Packet Detector:** this block detects IEEE 802.11ad compliant frames [33] received by the ADCs. It implements the Normalized Auto Correlation (NAC) algorithm given in Eq. (1), where $\mathbf{y}$ is the vector of received IQ samples, k is the sample index, and $[\cdot]^*$ is the complex conjugate operator.

$$\text{NAC} = \frac{\left|\sum_{n=k}^{k+127} [\mathbf{y}]_n [\mathbf{y}]^*_{n-127}\right|}{\left|\sum_{n=k}^{k+127} [\mathbf{y}]_n [\mathbf{y}]^*_n\right|} \tag{1}$$

In our implementation, the terms are rearranged to avoid division to simplify the hardware implementation of the block [7, 8, 17]. It considers that a valid frame is detected when the **NAC** exceeds a programmable threshold. To enhance the robustness and reduce the amount of false packet detection due to noise, we add a stage that counts the number of cycles during which the signal exceeds the configurable threshold. This reduces false frame detection but introduces additional latency in the detection. The block sets a flag (PD_FLAG) when a packet is detected. This flag can be used as trigger for other IP blocks. We use it in conjunction with the DDR write IP block to only write received IQ samples of actual frames to the DDR memory and avoid storing samples of noise. Packet detector configuration parameters, such as the threshold and the number of cycles to detect a valid packet, can be configured on-the-fly from the PS, making it flexible to adapt to different channel conditions without requiring a full re-build of the system.

• **Square-Root-Raised-Cosine (SRRC) filter:** this block is located in the receiver data path to process the incoming IQ samples from the ADC (Fig. 2b). The coefficients and the roll-off value of the filter can be configured from the PS considering that the IEEE 802.11ad standard does not define the baseband filter characteristics[2]. The filter includes a bypass option to disable its functionality.
• **GPIO signal management:** enables fast real-time beam steering of the antenna array from the PL by means of GPIO pulses. These are sent to the antenna controller (Fig. 2a) to perform the rapid beam pattern changes in the phased array. The block implements a state machine which continuously waits for an input trigger signal to generate the pulse to the RF front-end controller that changes the beam-pattern. The input trigger signal can be sent from the PS or from other hardware blocks in the PL. The block can either change a single beam pattern or can send multiple periodic pulses (user-configurable on a nanoseconds scale through the PS) from a single trigger pulse to cycle through multiple patterns. After sending the pulses, it can either directly return to the original idle state or generate another pulse (through a different GPIO pin) which forces the RF front-end to jump to a pre-configured standard beam pattern. This feature of mm-FLEX enables real-time experimentation not only with beam training mechanisms, including Sector Level Sweep (SLS) and BRP, where fast beam switching capabilities are used [33], but even supports custom switching capabilities on a *per symbol level*. We exploit this feature to implement the fast beam alignment algorithm proposed later in this paper.
• **Received Signal Strength (RSS) Computation:** estimates the RSS value of the received frame taking advantage of the term inside the summation in the denominator of (1), avoiding the use of complex multipliers that would unnecessarily increase the design's area. The block is configurable to allow any input trigger. In this paper we use the `PD_FLAG` as input trigger to start computing the RSS of a received frame. This block serves as a good example how to use mm-FLEX to extract characteristics of a received packet in real-time, allowing to quickly react to signal changes and avoid the possible latency when communicating with the PS.
• **Variable Delay Line:** the converters are connected to the FPGA using a JESD204B interface[4], which reduces the number of pins required for the connection. We calibrate the elastic buffers on the PL side of the JESD204B interface in order to reduce phase impairments between the IQ streams in the ADCs. This has to be done only a single time when a new FPGA image is built. For the DACs, this is accomplished by means of an IP block that implements a variable delay line which, at startup, is configured to read the phase registers on each DAC to compensate their possible differences using the IP. This step is required since the PLLs used to generate the clocks required by the JESD204B interface use a divided-by-4 version of the DAC sampling clock, which introduces phase uncertainly. With our approach, the phase difference between the IQ channels is ensured to be less than one sample. This block is transparent to the user and it is configured at startup once the user executes the "init" function (Section 2.2.2).

A block diagram of the baseband processor is presented in Fig. 2b, including the main FPGA hardware blocks described above. While Fig. 2b only includes the blocks corresponding to the Rx data path, the Tx data path is similar (replacing the ADCs by the DACs, the DDR write core by the DDR read core and replacing the receiver IP blocks by the ones used in the Tx chain). Furthermore, the system is capable of working as a full-duplex transceiver, since full-bandwidth IEEE 802.11ad compliant frames can be transmitted by the transmitter block independently of the receiver part of the system, streaming the IQ samples directly from the dedicated DDR memory to the Tx data path.

[2]Only Error Vector Magnitude (EVM) measurements are defined with an SRRC filter (roll-off = 0.25)

2.4 60 GHz RF front-end

We use the EVK06002 development kit from Sivers IMA[26] with 60 GHz up/down converters as the RF front-end for mm-FLEX. It supports the frequencies and bandwidth of IEEE 802.11ad channels and includes a 16+16 (Tx/Rx) element phased array arranged in two linear array structures (Fig. 2a). The kit includes 2D analog beam forming capabilities through phase shifters with 6 bit resolution for each antenna element. The kit can be controlled via three different interfaces: USB, SPI and GPIO pulses. We use the USB interface (connected to the AMC726 processor, Fig. 2a) to configure the kit as transmitter or receiver (at runtime), transfer the codebook, set the AWV and configure the GPIO control interface.

In addition, the kit incorporates fast beam switching capabilities through simple pulses via a GPIO interface, that allow to switch beam patterns every 10 *ns*. However, the kit has a maximum RF settling time of 35 *ns* when changing beam patterns. The GPIO functionality is as follows: i) one GPIO input is used to increment the pointer for the active sector in the stored codebook for each received pulse; ii) a second input is used to return to a predefined sector, which can be freely configured from the USB interface.

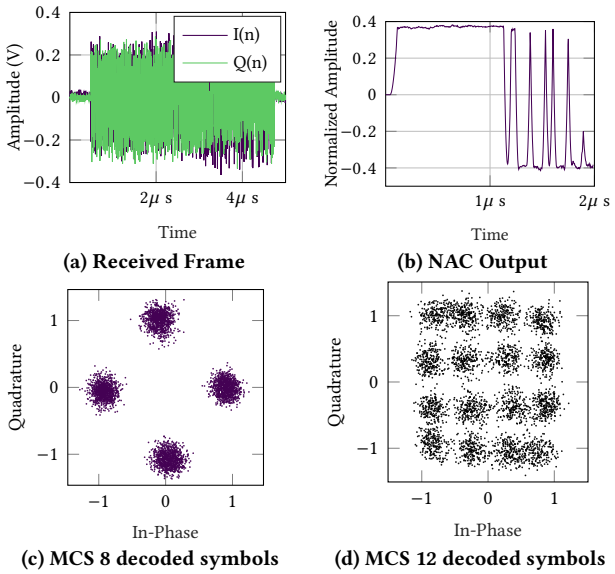
The front-end supports codebooks containing 64 different AWV. The predefined codebook consists of a quasi-omnidirectional beam pattern and 63 directional ones with main lobes at angles ranging from -45° to 45°. However, the codebook can be freely configured, allowing the user to design custom beam patterns as needed.

2.5 Preliminary test

To test the proper functioning of the developed system, we setup a 10 m Line-of-Sight (LOS) link and use mm-FLEX to transmit IEEE 802.11ad compliant frames (generated offline) for the different single-carrier Modulation and Coding Schemes (MCSs) of the standard, from MCS 1 to 12. The mm-FLEX receiver captures the frames after packet detection using the IP blocks described before, while the actual frame decoding is carried out in software. Fig. 3 shows an example of the experiment results. Specifically, Fig. 3a shows the signal of a received MCS 12 IEEE 802.11ad frame[33] over time and Fig. 3b shows the corresponding **NAC** output of the packet detector block. Finally, Fig. 3c and 3d show the constellation points for MCS 8 and MCS 12 frames of the corresponding frames captured using mm-FLEX.

3 CASE STUDY: ULTRA FAST BEAM ALIGNEMENT

To showcase the unique capabilities of mm-FLEX, we address the problem of high IEEE 802.11ad beam overhead [33] when the station (user) is mobile, especially for scenarios with a higher number



(a) Received Frame (b) NAC Output

(c) MCS 8 decoded symbols (d) MCS 12 decoded symbols

Figure 3: Example of received IEEE 802.11ad frames

of stations. Directional mm-wave communication requires proper beam alignment between the AP and stations to overcome the high path loss and achieve a sufficiently high SNR. The efficiency and accuracy of this mechanism directly determines the achievable data throughput. To improve performance, we introduce an Ultra Fast Alignment (UFA) mechanism, which aims to eliminate the beam training at the station side of the link by means of estimating the RSS for different receive beam patterns while the station is receiving the preamble of a training packet during AP beam training. This obviates the need for any dedicated station-side beam training, thus substantially reducing beam training overhead and maintaining beam alignment even when a station is highly mobile.

3.1 Preliminaries

Before discussing our algorithm, we introduce the beam training mechanism as defined by the IEEE 802.11ad standard and the preamble structure of standard compliant frames of which we take advantage in our algorithm.

3.1.1 IEEE 802.11ad beam training. APs perform beam training, called SLS, by sending beacon messages via each of their antenna sectors [14]. Stations are listening using a quasi-omnidirectional beam pattern. Then, stations perform their own SLS during the Association Beamforming Training (A-BFT) phase, by contending for one of the 8 A-BFT slots. If successful, they send sector sweep messages to the AP via their different antenna sectors. These messages also include the ID of the strongest AP sector. The AP then replies with a Sector Sweep Feedback that includes the ID of the strongest station sector, and the station concludes with a Sector Sweep ACK. Now, both AP and station are aware of the best sector to use. Stations can also perform their SLS in the Data Transfer Interval (DTI) instead of or in addition to the A-BFT. Whenever antenna sectors are (coarsely) aligned by means of the SLS, the devices can optionally carry out an additional BRP. This refinement phase allows to quickly probe several sectors over the course of a single packet and may be done with narrower antenna beams than during SLS to improve the link gain.

3.1.2 IEEE 802.11ad preamble processing. The 802.11ad frame includes a preamble composed of a Short Training Field (STF) and Channel Estimation Field (CEF), a header containing information about the MCS, data length, and other information, the data, and an optional Beamforming Training (BFT) field. The STF is used for frame detection, symbol synchronization, Carrier Frequency Offset (CFO) estimation and coarse synchronization, whereas the CEF is used to compute the Channel Impulse Response (CIR) to be used in the equalization blocks and for fine synchronization of the frame. The STF of control frames consist of 48 repetitions of Golay sequences of length 128 Gb_{128}, followed by a single $-Gb_{128}$ and $-Ga_{128}$ sequence to detect the end of the STF [33]. The STF duration is 3.63 μs.

3.2 Main idea

The purpose of UFA is to reduce beam training overhead by performing SLS only at the AP and train the stations while they are receiving the AP transmit sector training packets. We first discuss the case of a single AP performing SLS with a station, but this case easily extends to multiple APs.

During the AP's SLS, the station receives a beacon frame for each AP transmit beam pattern. After the station successfully detects a beacon frame, the STF has served its purpose and for the remainder of its duration the station can switch through M beam patterns. The switching has to terminate before the STF ends, i.e., before the Golay sequences $-Gb_{128}$ and $-Ga_{128}$ at the end of the STF, to ensure that boundary detection is not affected and the CEF can be received correctly.

The key idea of our algorithm is that since for a given transmit sector the M beam pattern changes of the station happen during a single STF, changes in the amplitude of the signal are directly related to the receive beam patterns. These amplitude changes can be extracted from the RSS measurements, which provides a simple and fast estimator that does not increase receiver complexity. We then use the sparsity of the mm-wave channel to estimate the AoA with high accuracy with only a reduced set of measurements.

3.3 Problem formulation

For a given transmit beam pattern t and a receive beam pattern r, the $\text{RSS}_{t,r}$ can be calculated as

$$\text{RSS}_{t,r} = \sum_{n=k}^{N-1+k} [y]_n \cdot [y]_n^* \tag{2}$$

where y is the vector with the received IQ samples, k denotes the sample index and N is the number of IQ samples considered to compute the RSS value. We denote the Euclidean norm by $\|\mathbf{y}\|$ and the inner product of two vectors x and y by $< x, y >$. For a single transmit sector t, the station changes through M receive sectors.

Let $R_t = [\text{RSS}_{t,1}, \ldots, \text{RSS}_{t,M}]$ be the vector containing the M different RSS measurements. For the receive beam patterns, let G denote the size of the grid of the possible angles for which the antenna gain is known. Therefore, $b_m = [b_m(\theta_1), \ldots, b_m(\theta_G)]$ defines the

shape of the m^{th} receive beam pattern and $B = [B(\theta_1), \ldots, B(\theta_G)]^T$ is the matrix with columns containing the measurement of $b_m(\theta_i)$ $\forall m$. In order to obtain the AoA θ at the station, we must find the θ value that minimizes the differences between the $\text{RSS}_{t,r}$ measurements at the receiver and the known receive beam patterns b_m over all transmit sectors

$$\theta = \arg\min_{\theta} \sum_{t,m} \min_{\alpha_t(\theta)} \|\text{RSS}_{t,m} - \alpha_t(\theta) b_m(\theta)\|^2, \tag{3}$$

where $\alpha_t(\theta)$ stands for the amplitude of the received beam pattern t that minimizes the Euclidean norm for each t. We convert this problem to a correlation between the RSS measurements and the known beam patterns. For this, we expand Eq. (3) to obtain:

$$\theta = \arg\min_{\theta} \sum_{t,m} \min_{\alpha_t(\theta)} \|\text{RSS}_{t,m}\|^2 + \|\alpha_t(\theta) b_m(\theta)\|^2 - 2\alpha_t(\theta)\text{RSS}_{t,m} b_m(\theta)$$

Summing over m we get

$$\theta = \arg\min_{\theta} \sum_{t} \min_{\alpha_t(\theta)} \|R_t\|^2 + \|\alpha_t(\theta) B(\theta)\|^2 - 2\alpha_t(\theta) < R_t, B(\theta) >$$

Now, the minimization over $\alpha_t(\theta)$ is a quadratic problem that we can solve and substitute while removing the constant term

$$\theta = \arg\min_{\theta} \sum_{t} - < R_t, B(\theta) >^2 / \|B(\theta)\|^2.$$

Finally we change the variable $\bar{B}(\theta) = B(\theta)/\|B(\theta)\|$ and convert the problem into a maximization problem

$$\theta = \arg\max_{\theta} \sum_{t} < R_t, \bar{B}(\theta) >^2 . \tag{4}$$

The station uses the estimated direction of arrival θ of the signal to select the beam pattern with the highest gain in that direction for communication. This method is of very low complexity and uses the RSS information from the M receive sectors for all the received transmit sectors during SLS, in a single optimization. All steps of our algorithm are basic operations such as additions and real multiplications that allow for fast hardware implementation and low complexity at the receiver.

Finally, it is important to remark that the accuracy of mm-FLEX does not rely on the shape of the M beam patterns used to estimate θ but on the angular resolution used to measure them. In the evaluation section, we will show that very accurate angle estimation is possible measuring only a small subset of beam patterns from the 63 predefined ones. For the communication phase, we then choose the best matching pattern from all of the 63 predefined ones. To further improve performance, it would even be possible to design a custom beam pattern that has the highest gain at the desired angle.

4 IMPLEMENTATION

We now explain how the mm-FLEX testbed is configured to implement our beam-alignment algorithm, including the enhanced functionality added to the IP blocks presented in Section 2.

4.1 mm-FLEX Configuration

The UFA algorithm is implemented using the mm-FLEX IP blocks explained in Section 2. We introduce some modifications from the standard functionality to provide rapid antenna reconfiguration and the *synchronized* computation of multiple RSS measurements that are required to implement the algorithm.

The *packet detector* generates a trigger signal (`PD_FLAG`) that is used as input for the GPIO signal management and the DDR write block. The latter allows to save to the DDR memory only valid data corresponding to the beacon frames sent from the multiple APs in the experimental setup. The block is configured with a detection threshold of 0.25 and the number of clock cycles to detect a valid packet is set to 64. These parameters reduce the probability of false packet detection while maintaining a reasonable detection latency.

The *GPIO signal management* block is configured to start sending the GPIO pulses to change through M beam patterns after receiving the `PD_FLAG` as input trigger. In the beginning, the system starts with a quasi-omnidirectional beam pattern as conventional IEEE 802.11ad receivers. Once it detects a valid frame, it triggers pulses to cycle through M measurement beam patterns from the predefined codebook. After that, another pulse is sent to return to the quasi-omnidirectional beam pattern to continue with the frame decoding. The time between GPIO pulses is selected to establish a balance between RSS accuracy and computation time. To this end, the IP block is configured to trigger GPIO pulses each 192 received samples (12 FPGA clock cycles = 54.5 *ns*). Although the M value is user configurable from the PS, for the experiments conducted in this paper we set $M = 10$, which gives (192 samples * 10)/3.52 GSPS = 545 *ns* for the beam pattern changes.

We augment the functionality of the *RSS computation* block to be synchronized with the GPIO signal management block to compute M RSS values per detected packet, one per tested receive beam pattern. The block waits for a `PD_FLAG` to start computing each RSS value during 12 clock cycles (leaving out the first 8 samples of the RSS computation to avoid the possible transients in the received signal due to the RF front-end settling time of 35 *ns* when changing a beam pattern) and saves the individual results to M registers to build the vector R_t used to compute the AoA estimate.

We remark that there exists a constant delay between the time when the ADCs sample the incoming IQ signal and when the corresponding samples are available to be processed by the FPGA logic. Due to the buffering and serialization process in the JESD204B interface[4], there is a deterministic latency of 650 *ns* (measured) for both ADCs in the system. This value increases to 700 *ns* due to the pipeline registers used in our SRRC filter and packet detector blocks to meet timing constraints in the design. The RSS computation block takes this deterministic latency into account before it starts computing the RSS values for each beam pattern. In Fig. 4 we present a timing diagram showing how the RSS computation block works cooperatively with the GPIO signal management and packet detector blocks to compute the M RSS values (shown in different colours), when a valid packet is detected.

In Fig. 5, we show an example of a preamble with $M = 10$ beam pattern changes (separated with dotted lines). We observe a typical frame detection delay of 900 ns, a latency of 700 ns between the Analog-to-Digital Converter (ADC) and FPGA (before the FPGA

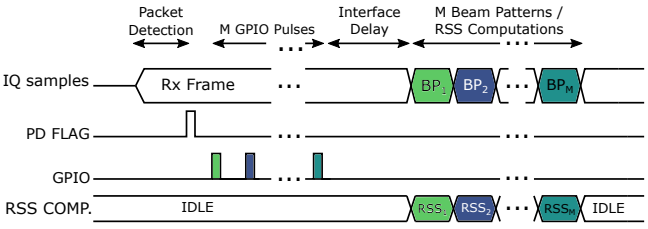


Figure 4: Example Timing Diagram

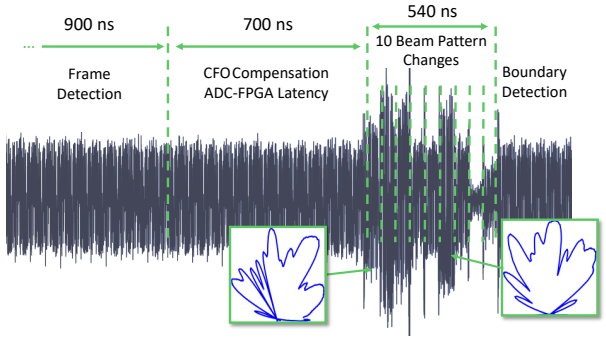
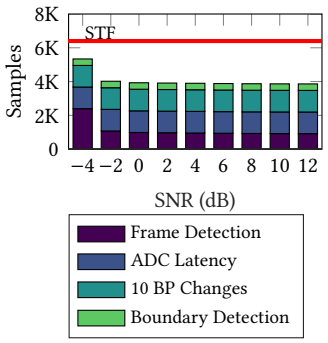


Figure 5: Measured STF of a control frame, showing signal changes due to beam switching at the receiver

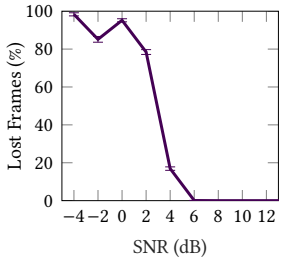
can react to the GPIO pulses after the detection and starts seeing the beam patterns changes). During this time we also perform CFO estimation (which requires around 7 Golay sequences to obtain a good estimate [7, 8]) and symbol synchronization, and 545 ns to switch through the 10 beam patterns, which is more than enough to reliably estimate the AoA of the received signal. The total time is well below the is 3.49 μs duration of the 48 Golay sequences Gb_{128} of the STF. As can be seen from Figure 5, switching beam patterns is extremely fast and only short transients can be observed close to each beam pattern change, affecting only few IQ samples during the phase shifter reconfiguration. Symbol synchronization aims to select the proper sampling point of the signal, splitting it in a small set of coarse phases. After that, each one is correlated with Golay sequences and then the one with the maximum value is selected. This can be done while the decoder is waiting for the changes in the signal beam pattern to take effect.

4.2 Measuring the receive beam patterns

UFA requires knowing the shape of the M beam patterns used at the receiver to estimate the AoA. In order to measure the shape of the receive beam patterns, we mount the phased array on a motorized pan/tilt platform that mechanically steers over the whole azimuth (Fig. 2a). We time the movement of the rotation table to wait for a full SLS before moving to the next angle. We record $n = 10$ measurements of the received signal for each receive beam pattern r and transmit beam pattern t, respectively, and then change the angle for a new measurement. The rotation table advances in steps of 0.5°, giving approximately 0.5° resolution in our direction estimation. With the obtained measurements, we average the individual RSS measurements for each one of the angles of the pan/tilt platform to obtain the amplitude response of the receive beam patterns. Note



(a) Preamble processing time



(b) Fraction of undecodable SLS frames

Figure 6: Frame detection for different SNR values

that we only need to take these calibration measurements once per type of device and they are independent of the deployment scenario. The measurements are stored in a ROM on the FPGA.

5 EVALUATION

We now evaluate the performance of our mm-FLEX beam-alignment algorithm. Unless stated otherwise, we set up the experiments in an empty room of size 6 × 23 m, where reflections mainly come from the walls. To show that mm-FLEX works with Commercial-Off-The-Shelf (COTS) APs, we use mm-FLEX as the station and a commercial Talon AD7200 router as the AP. The Talon router uses Qualcomm's QCA9500 FullMAC IEEE 802.11ad Wi-Fi chip and a phased array antenna with 32 elements. We also perform tests with a Dell Wireless Dock D5000 station and a NETGEAR Nighthawk X10 router.

5.1 Frame detection latency

UFA critically depends on timely frame detection during preamble reception. If frame detection occurs late in the preamble, there may not be enough time to switch through directional receive beam patterns and back to the quasi-omnidirectional pattern before the end of the STF, which would compromise packet reception. Detection time primarily depends on the SNR. We therefore change the receiver gain to obtain different SNR values. Receive beam pattern switching at the station is disabled, to determine the exact remaining duration of the STF after the moment of frame detection.

In Fig. 6a we stack the delay for frame detection, the constant ADC-to-FPGA latency, the time required for 10 receive beam pattern changes, as well as the time for boundary detection. We also include a line at 6400 samples corresponding to the duration of the STF. We observe that for SNR values from 14 to -4 dB the frame detection delay indeed increases, but for all SNR values packet detection occurs early enough for our proposal to work. Moreover, as shown in Fig. 6b, for SNR values below 2 dB, most of the frames cannot be correctly decoded. SLS frames are control frames that have a more robust preamble than data frames, so that data communication at these SNR levels would be impossible. UFA thus works for all relevant SNR levels.

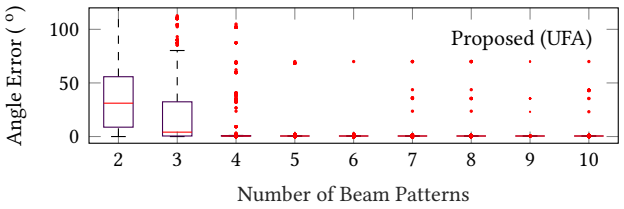


Figure 7: Angle error for different reception beam patterns

5.2 Accuracy

We study the impact of the number of receive beam patterns during the preamble and the number of sectors in the SLS on the angle estimation accuracy, obtained over 920 measurements each, for a single AP and one station. The latter is rotated between −45° and 45° in steps of 2°. Fig. 7 shows whisker plots with median, quartiles, extremes (± 1.5 IQR), and outliers. Despite the possible multi-path effect, our method is able to determine the direction of the AP with a mean error of 0.5° with only 5 beam patterns (with a very small number of outliers).

Fig. 8 shows the Cumulative Distribution Function (CDF) of the AoA estimation error, obtained with UFA for 3, 5 and 10 receive beam patterns. We show the case where we only consider a subset of transmit sectors (14 instead of all possible sectors in the SLS). For UFA-10BP, the CDF curve is well below 5° for 97% of the cases and thus within the half power beam width of the antenna (~10°). The algorithm achieves less than 4° error for 95% of the cases and a median angle estimation error of 2°. The performance of UFA-5BP is similar to the one of UFA-10BP without introducing any relevant accuracy loss. If we consider only three receive beam patterns, the selected beam pattern for communication would be sub-optimal as shown in the UFA-3BP curve. For UFA-5BP, the level of accuracy is more than sufficient for good beam alignment performance.

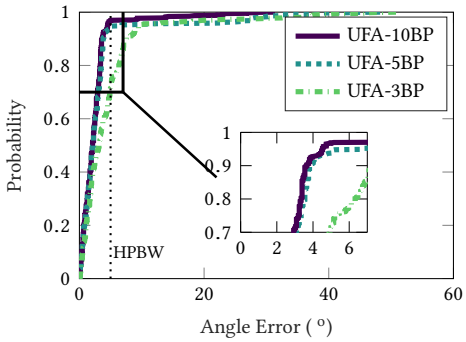


Figure 8: CDF of the AoA estimation error for an AP with 14 sectors

5.3 SNR loss

Whenever the estimated angle leads to the selection of a sub-optimal beam pattern, the SNR of the link will be lower by factor that corresponds to difference of the gains of the optimum versus the chosen antenna pattern at the true angle. We compare the SNR loss of UFA with 10 receive beam patterns to brute force beam training which always selects the optimal beam pattern. To determine SNR, we generate traffic from the AP to the station using *iperf*. After AoA estimation using UFA, we configure the station to receive the data packets using the beam pattern that provides the highest gain in the estimated direction. We then decode the data packets from the AP to determine the SNR.[3]

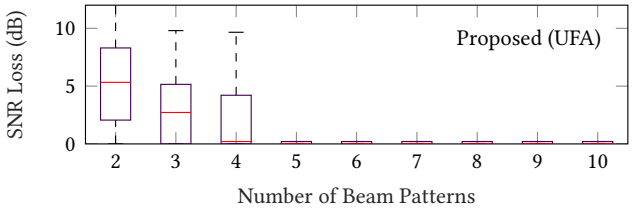


Figure 9: SNR-loss of UFA compared to exhaustive search

The SNR loss of UFA compared to exhaustive search is shown in Fig. 9. The mean SNR loss is close to zero for more than 3 receive beam patterns, and has no non-zero outliers for 5 beam patterns or more.

5.4 Mobility

To verify the angle estimation accuracy of UFA in real-time, we perform tests with rotation at a fixed location and trajectories through the room.

5.4.1 Rotation test I. First we use a setup where the mm-FLEX station is static but rotates on a pan/tilt platform, to be able to tightly control the angles. The receiver is steered following a triangular angle pattern sweeping to angles of 35° and -35° and back to 0°. The complete cycle of the movement takes 2.8 seconds. After that, the system remains with the original orientation (pointing towards AP1). For this experiment we use 10 receive beam pattern changes. We deploy 5 APs in the room to show that UFA can reliably estimate angles to multiple APs at the same time, as pictured in Fig. 10. Furthermore, to show that UFA works with any standard-compliant IEEE 802.11ad hardware, we use different types of APs: one Talon AD7200 router (AP1), 3 Dell Wireless Dock D5000 stations (AP2, AP3, AP5), and a NETGEAR Nighthawk X10 (AP4). The distances to the station are: AP1 = 6 m, AP2 = 6.5 m, AP3 = 4.7 m, AP4 = 6.5 m and AP5 = 4.7 m. The results of this experiment are presented in Fig. 10. The initial ground truth angles for each AP are AP1 = 0°, AP2 = 21.8°, AP3 = 30.9°, AP4 = −21.8°, AP5 = −30.9°. As expected, the figure shows the triangular angle rotation pattern and flat lines once rotation ends after 2.8 seconds. The rotation is relatively fast, corresponding to angle changes of 55° per second. We further include in Fig. 10 the absolute angle error over time for the 5 APs. The median angle estimation errors for the APs are 1.17°, 0.83°, 0.99°, 0.63°, and 1.21°, respectively. Given that our pan/tilt system and mm-FLEX are not perfectly time synchronized and a 1° error corresponds to only a 18 ms time shift, these results are within the measurement accuracy of our setup.

[3]Since our FPGA implementation does not include real-time frame decoding, the ACK frames for the data packets from the AP are sent by a Talon router in station mode located next to the FPGA.

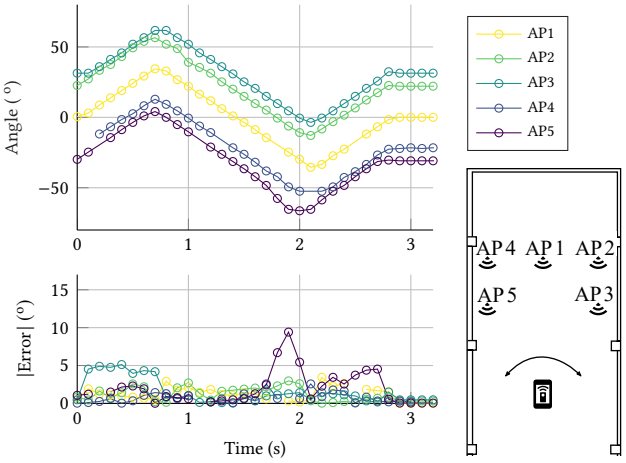


Figure 10: Real-time rotation of the station with 5 APs at different positions in the room (left) and experimental setup for rotation experiment (right)

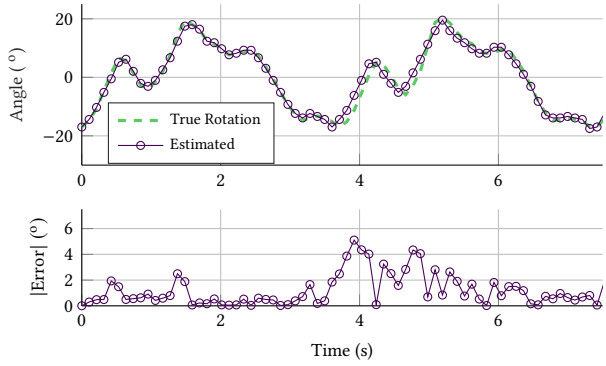


Figure 11: Real-time rotation of the station following the mobility trace of a person walking

5.4.2 Rotation test II. We perform an additional rotation test in an smaller scenario, a laboratory of 7×6 m, where windows and office furniture (chairs, desks, computers, etc.) create a richer multi-path environment. We deploy a single AP-station pair spaced 4 m apart and facing each other. The station is rotating following a pattern given by a gyroscope trace of a walking person [10, 25], while the AP is in a fixed position. The results of the experiment are presented in Fig. 11, showing how the station implementing UFA is able to follow the trajectory over the duration of the experiments with a maximum absolute error of 5.1° and a median error of 0.69°, despite the more complex multi-path environment in the laboratory.

5.4.3 Human mobility I. We also perform experiments with real-time human mobility in which the receiver is moved manually at walking speed along a trajectory within the room with four APs placed in the corners, as depicted in Fig. 12. The APs are performing SLS one at a time every 102 ms. Because of the different beam width apertures and the orientations of the receiver at each point, different APs are observed during different parts of the trajectory, as shown in Fig. 12. Obtaining ground truth angles for this setup is

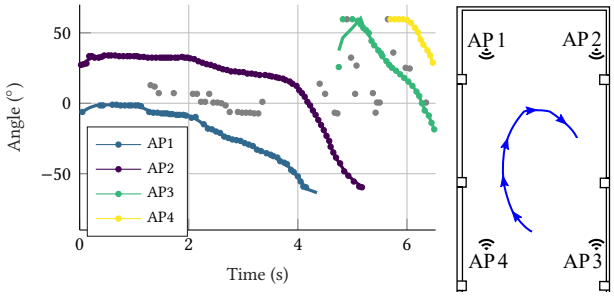


Figure 12: Angle of estimated LOS path for mobility experiment with 4 APs (left) and experimental setup for mobility experiment (right)

extremely difficult since our manual angle measurements as well as those of an electronic compass devices have an angle error that is higher than the accuracy of our algorithm. While we could not compare against a sufficiently accurate ground truth, we expect the data to be smooth due to the continuity of the movement. We smooth the angle trajectory of each APs using a 10 sample window Savitzky-Golay filter and compare against it as we would compare against ground truth. This does not provide an absolute error but a relative error that indicates how far angle estimates stray from the smooth trajectory. The obtained errors for the measured paths are very accurate with 0.36°, 0.22°, 0.45°, and 0.76° for the respective APs. In addition, we observe that the angles change in the order we expect for the route followed, based on our manual angle estimation. This indicates that our results well match the angles observed throughout the experiment trajectory.

To avoid spurious estimates to influence the trajectory results of Fig. 12 due to multipath or bad quality signal, we apply the following condition as a filter: a minimum ratio p_{th} of the observed power must be explained by the estimated direction expected measure, else we discard the result, i.e., $\mathbb{E}_t \|\alpha_t B(\theta)\|^2 / \|R_t\| > p_{\text{th}}$. Following the previous derivations can be shown to correspond to

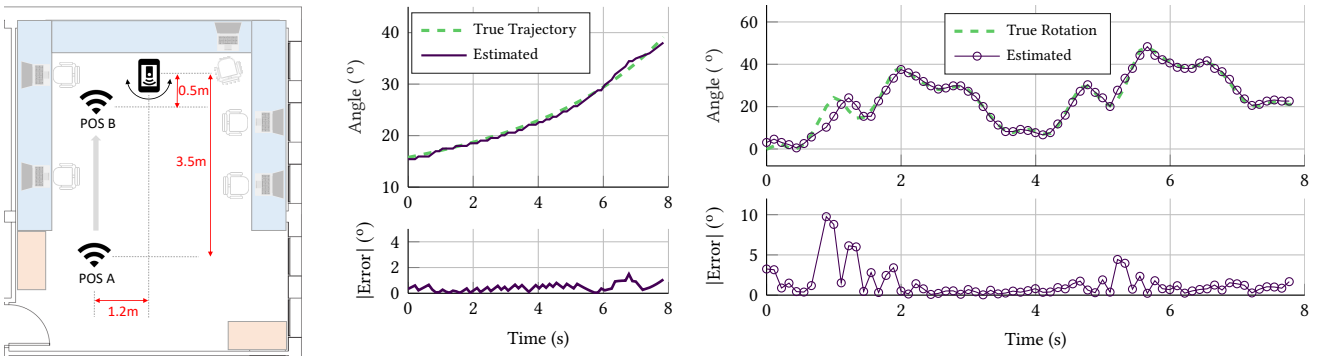
$$\mathbb{E}_t | < R_t, B(\theta) > | / \|R_t\| > p_{\text{th}}.$$

For our experiments this value is set to $p_{\text{th}} = 0.85$.

5.4.4 Human mobility II. We also perform an experiment in the smaller laboratory environment with a single AP-station pair, as shown in Fig. 13a. First, we take measurements with the *AP moving* from POS A to POS B at constant walking speed with the station in a fixed position. The result from Fig. 13b shows that UFA is able to estimate the trajectory with high accuracy and a median error of 0.39°. Next, we test the system with both AP and station mobility: while the station is rotating following the human walking pattern trace [10, 25], the AP is moving from POS A to POS B as in the experiment from Fig. 13b. The results in Fig. 13c show that the angle pattern is the combination of the angles in Fig. 13b and Fig. 11. Even with the difficulties derived from the dual movement, the error curve shows a maximum error of 10° and a median of 0.78°.

5.5 Scalability and overhead reduction

While we only have two FPGA transceivers, since UFA is passive and simply removes station beam training overhead entirely, we



(a) Laboratory Environment Setup (b) Vertical Trajectory (c) Dual AP / station movement

Figure 13: Both AP *and* station mobility in a laboratory environment

can reliably estimate the overhead reduction by analyzing large standard IEEE 802.11ad deployments and discounting the station overhead. We test real-world deployments with up to 10 Talon routers, out of which 1 is configured as AP, 7 are stations, and 2 are in monitor mode to capture the IEEE 802.11ad traffic. The beam training frequency is vendor specific and in the Talon routers depends on the data traffic and medium utilization. For this reason we add background traffic of 250 MBit/s per station and compare actual measured overhead and extrapolated overhead. The extrapolated overhead is determined using the per station overhead (Sector Sweep during A-BFT and DTI, Sector Sweep Feedback, and Sector Sweep ACK) measured with 1 AP and 1 station, multiplied by the number of stations. This corresponds to deployments where the frequency of beam training cannot be reduced due to network dynamics. To compute the overhead reduction achieved by UFA, we divide the total overhead by the overhead caused by UFA (DMG Beacon, Sector Sweep Feedback, and Sector Sweep ACK).

Fig. 14 shows measured and extrapolated beam training overhead. With 1 AP and 1 station, their overhead is approximately the same and thus IEEE 802.11ad has twice the overhead of UFA. For 4 stations, the extrapolated station overhead is 4 times larger and thus IEEE 802.11ad overhead is 5.1× that of UFA (left bar). The Talon routers actually both slightly increase the DMG Beacon frequency and significantly increase the station beam training frequency so that measured IEEE 802.11ad overhead is higher by a factor of 5.9. For 7 stations, the beam training frequency of stations is reduced due to the increase in data traffic with that many stations. Here, IEEE 802.11ad overhead is around 8× that of UFA, and measured and extrapolated overhead almost match.

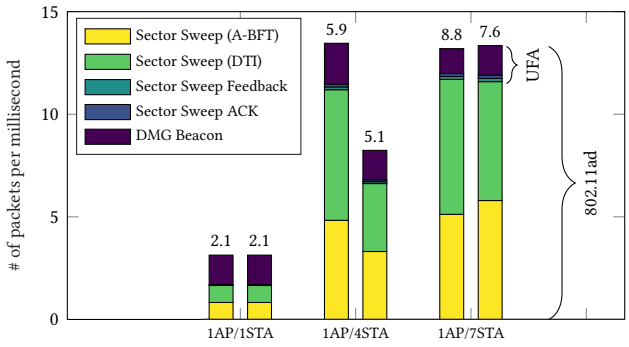


Figure 14: Measured (left bar) and extrapolated (right bar) overhead of IEEE 802.11ad and UFA, with the overhead ratio of IEEE 802.11ad divided by UFA shown on top of the bars

6 RELATED WORK

We first discuss works related to mm-wave testbeds and also briefly survey research on beam training improvements.

mm-wave testbeds: Since the advent of mm-wave communication systems, different experimentation platforms have been proposed, each one trying to cover different types of experiments. Solutions differ in price, performance, flexibility, etc. The MiRa platform [2] uses the same baseband equipment typically used for sub-6GHz platforms, simple USRP software-defined radio devices, attached to a custom-made RF front-end in the 28GHz frequency band. Due to the bandwidth constraints of the USRP baseband processor, only narrowband signal can be generated, which does not allow to study the characteristics of ultra wideband mm-wave channels. Besides, the signal processing functions are implemented in software which introduces a high latency and makes such a testbed unsuitable for certain applications. The phased array provides only coarse phase control, limiting its application for relatively simple directional applications. Despite that, its low cost makes it a good solution for simple research experiments. Recently, the solution from [13], use similar equipment as [2] but introduces the capability of implementing signal processing functions on the on-board FPGA. This reduces the latency and allows for the validation of hardware implemented algorithms. Despite that, it suffers from the similar disadvantages as [2]. The X60 testbed [23] in turn addresses the need for high-end full bandwidth systems and includes hardware equipment from National Instruments (NI) plus a phased array antenna kit with analog beam forming capabilities. X60 fulfills the bandwidth requirements of the IEEE 802.11ad standard, but due to the complex hardware has a comparatively high cost that makes it unaffordable for much of the research community. Furthermore, it has limited flexibility regarding the fast configuration of the antenna array. OpenMili from [35] increases the bandwidth capabilities compared to [2, 13], but still does not meet the requirements of mm-wave standards,

such as the IEEE 802.11ad[33]. Besides the custom-made phased array system only includes coarse phase control and its antenna control is limited.

COTS devices have been used to perform mm-wave experiments. While this offers the cheapest solution, this approach does not provide access to the physical and lower MAC layers, which are important to develop and validate new MAC and signal processing solutions. The work from [29] modifies the firmware of a COTS device to allow access to some of the physical layer information (SNR, RSS, etc.), providing the basis for a variety of research efforts [11, 15, 16].

M-Cube [36] makes use of an RF front-end with phased array antennas from a COTS device, which are connected to an FPGA-based baseband processor through a custom-made bridge board. The system sends the adequate control signals (acquired by means of reverse engineering) from the baseband processor to reconfigure the RF front-end. As drawbacks, the latency required to reconfigure the antenna array may be too high for certain applications such as the processing of TRN field of IEEE 802.11ad/ay frames. Besides, the separation of data and control paths via different devices reduces the flexibility of the testbed for dynamic environments. A very promising testbed design is to use mm-FLEX with the bridge board and RF front-end from [36] and which allows to have both control and data paths in the same device.

With mm-FLEX, we target the next milestone for mm-wave experimentation, mobility. The fast real-time antenna reconfiguration makes mm-FLEX an excellent choice for mobility experiments where fast beam-steering is important. Besides, it offers the flexibility to be used not only for IEEE 802.11ad-based systems but also with other mm-wave standards such as the IEEE 802.11ay, where fast beam steering is widely used during the beam training mechanism and also to track station movement in the data communication phase.

To summarize, in Table 1, we include a comparison between mm-FLEX and different mm-wave testbeds from the literature. The comparison includes cost, operating bandwidth (BW), easiness to upgrade the system functionality, antenna type included in the RF front-end, number of antenna elements and fast beam steering capability.

Table 1: Comparison of mm-FLEX and existing mm-wave testbeds

	mm-FLEX	Open-Mili [35]	RFNoC-based [13]	X60 [23]
COST	**40K**	29K	24K	200K
BW	**2GHz**	1GHz	200MHz	2GHz
UPGRADING	**YES**	LIMITED	LIMITED	DIFFICULT
ANT. TYPE	**ARRAY**	ARRAY	HORN	ARRAY
ANT. ELEM.	**16+16**	4	1	12
FAST BEAM STEERING	**YES**	NO	NO	NO

Beam training: Beam training using the IEEE 802.11ad standard mechanism takes several milliseconds [15, 30, 37]. This may incur unacceptable latency and a very high overhead in mobile scenarios where beam training has to be performed frequently [1, 9, 21]. Mechanisms that reduce this overhead are mainly based on Compressed Sensing (CS) beam training, since the sparsity of the mm-wave channel makes it amenable to sparse estimation [19, 28]. Recent approaches along this line [3, 20, 29] use RSS measurements to obtain angle information. However, they only take advantage of changing beam patterns at one side of the link, requiring three times more measurements for the same accuracy (15 instead of 5) and explicit station beam training, whereas our approach does not require per client beam training. In [29], both SNR and RSS measurements are used due to the noisy RSS values extracted from the COTS devices used for their experiments. Other works present simultaneous station training in 802.11ad [18, 24], but none of those are compatible with legacy IEEE 802.11ad devices. A 2-step location algorithm is proposed in [12] were AP and STA share information to try to overcome the imperfect position problem, however this work is only simulation based and the overhead is unclear. In [5], the authors extract the association directions from the A-BFT, but their focus is on maximizing the number of concurrent paths whereas the application we showcase aims to reduce the beam training overhead. We provide better mobility support by eliminating station beam training whereas prior work requires active station beam training. Our approach is compatible with current 802.11ad deployments, does not affect legacy stations and can be implemented on current devices using only firmware modifications.

7 CONCLUSIONS AND REMARKS

In this paper we propose mm-FLEX, a flexible, modular and reconfigurable testbed suitable for experiments in mm-wave ultra wideband channels. Our proposal allows easy integration and verification of hardware processing blocks as well as real-time control of phased array antennas for experiments in high mobility scenarios. The open-source nature of our mm-wave system makes it an ideal platform for the research community to modify and extend the functionality as needed. The design is powerful enough to allow for complex low level changes to the mm-wave signal processing, and at the same time offers a simple ready-to-use platform for researchers that want to focus on the higher layers of the protocol stack, using the physical layer implementation as is. We test the most important features of mm-FLEX by implementing a mechanism for fast IEEE 802.11ad beam training. To this end, we make use of the powerful beam pattern switching capabilities of mm-FLEX together with fast RSS measurements. Our evaluation shows that mm-FLEX provides extremely accurate angle measurements at the station that indicate the best station beam pattern to be used, using only a reduced set of measurement during a single preamble of a received frame. This obviates the need for active station beam training and thus significantly reduces the beam training overhead for dense and mobile networks.

ACKNOWLEDGMENT

This work was supported in part by the Region of Madrid through TAPIR-CM (S2018/TCS-4496) and PinPoint5G+ RTI2018-094313-B-I00 (MCIU/AEI/FEDER, UE).

REFERENCES

[1] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. 2016. Cutting the Cord in Virtual Reality. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets '16)*. Association for Computing Machinery, New York, NY, USA, 162–168. https://doi.org/10.1145/3005745.3005770

[2] Omid Abari, Haitham Hassanieh, Michael Rodriguez, and Dina Katabi. 2016. Poster: A Millimeter Wave Software Defined Radio Platform with Phased Arrays. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 419–420. https://doi.org/10.1145/2973750.2985258

[3] Haitham Hassanieh, Omid Abari, Michael Rodriguez, Mohammed Abdelghany, Dina Katabi, and Piotr Indyk. 2018. Fast Millimeter Wave Beam Alignment. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM '18)*. Association for Computing Machinery, New York, NY, USA, 432–445. https://doi.org/10.1145/3230543.3230581

[4] JEDEC Solid State Technology Association. 2017. Serial Interface for Data Converters (JESD204B). (2017). http://www.jedec.org/standards-documents/results/jesd204

[5] Suraj Jog, Jiaming Wang, Junfeng Guan, Thomas Moon, Haitham Hassanieh, and Romit Roy Choudhury. 2019. Many-to-Many Beam Alignment in Millimeter Wave Networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. USENIX Association, Boston, MA, 783–800. https://www.usenix.org/conference/nsdi19/presentation/jog

[6] Jesus Omar Lacruz and Joerg Widmer. 2020. mm-FLEX: Millimeter Wave Mobile Full-Bandwidth Experimentation Platform. (2020). http://wireless.networks.imdea.org/mmflex

[7] Chun-Yi Liu, Meng-Siou Sie, Edmund W. J. Leong, Yu-Cheng Yao, Chih-Wei Jen, Wei-Chang Liu, Chih-Feng Wu, and Shyh-Jye Jou. 2017. Dual-Mode All-Digital Baseband Receiver With a Feed-Forward and Shared-Memory Architecture for Dual-Standard Over 60 GHz NLOS Channel. *IEEE Transactions on Circuits and Systems I: Regular Papers* (2017).

[8] Wei-Chang Liu, Ting-Chen Wei, Ya-Shiue Huang, Ching-Da Chan, and Shyh-Jye Jou. 2015. All-Digital Synchronization for SC/OFDM Mode of IEEE 802.15.3c and IEEE 802.11ad. *IEEE Transactions on Circuits and Systems I: Regular Papers* (2015).

[9] A. Loch, A. Asadi, G. H. Sim, J. Widmer, and M. Hollick. 2017. mm-Wave on wheels: Practical 60 GHz vehicular communication without beam training. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*. 1–8.

[10] Adrian Loch, Hany Assasa, Joan Palacios, Joerg Widmer, Hans Suys, and Björn Debaillie. 2017. Zero Overhead Device Tracking in 60 GHz Wireless Networks using Multi-Lobe Beam Patterns. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '17)*. Association for Computing Machinery, New York, NY, USA, 224–237. https://doi.org/10.1145/3143361.3143395

[11] Adrian Loch, Guillermo Bielsa, and Joerg Widmer. 2016. Practical Lower Layer 60 GHz Measurements Using Commercial Off-the-shelf Hardware. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization (WiNTECH '16)*. ACM, New York, NY, USA, 9–16.

[12] F. Maschietti, D. Gesbert, P. de Kerret, and H. Wymeersch. 2017. Robust Location-Aided Beam Alignment in Millimeter Wave Massive MIMO. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–6.

[13] Adriana Moreno, Jesus O. Lacruz, and Joerg Widmer. 2020. Open Source RFNoC-Based Testbed for Millimeter-Wave Experimentation Using USRP Software Defined Radios. In *International Symposium on Circuits and Systems (ISCAS-2020)*.

[14] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer. 2014. IEEE 802.11ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi [Invited Paper]. *IEEE Communications Magazine* 52, 12 (2014), 132–141.

[15] J. Palacios, P. Casari, H. Assasa, and J. Widmer. 2019. LEAP: Location Estimation and Predictive Handover with Consumer-Grade mmWave Devices. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. 2377–2385.

[16] Joan Palacios, Daniel Steinmetzer, Adrian Loch, Matthias Hollick, and Joerg Widmer. 2018. Adaptive Codebook Optimization for Beam Training on Off-the-Shelf IEEE 802.11Ad Devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. ACM, New York, NY, USA, 241–255.

[17] N. Preyss and A. Burg. 2015. Digital synchronization for symbol-spaced IEEE802.11ad Gigabit mmWave systems. In *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*. 637–640.

[18] J. Qiao, X. Shen, J. W. Mark, and Y. He. 2015. MAC-layer concurrent beamforming protocol for indoor millimeter-wave networks. *IEEE Transactions on Vehicular Technology* 64, 1 (2015), 327–338.

[19] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir. 2013. Broadband Millimeter-Wave Propagation Measurements and Models Using Adaptive-Beam Antennas for Outdoor Urban Cellular Communications. *IEEE Transactions on Antennas and Propagation* 61, 4 (2013), 1850–1859.

[20] Maryam Eslami Rasekh, Zhinus Marzi, Yanzi Zhu, Upamanyu Madhow, and Haitao Zheng. 2017. Noncoherent MmWave Path Tracking. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications (HotMobile '17)*. Association for Computing Machinery, New York, NY, USA, 13–18. https://doi.org/10.1145/3032970.3032974

[21] Robert W. Heath. 2015. Vehicular Millimeter Wave Communications: Opportunities and Challenges. (2015). http://users.ece.utexas.edu/~rheath/presentations/2015/DSTOPvehicularMmWave2015Heath.pdf

[22] W. Roh, J. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar. 2014. Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results. *IEEE Communications Magazine* 52, 2 (2014), 106–113.

[23] Swetank Kumar Saha, Yasaman Ghasempour, Muhammad Kumail Haider, Tariq Siddiqui, Paulo De Melo, Neerad Somanchi, Luke Zakrajsek, Arjun Singh, Roshan Shyamsunder, Owen Torres, et al. 2019. X60: A programmable testbed for wideband 60 GHz WLANs with phased arrays. *Computer Communications* 133 (2019), 77 – 88. https://doi.org/10.1016/j.comcom.2018.09.005

[24] S. Shao, H. Zhang, D. Koutsonikolas, and A. Khreishah. 2018. Two-dimensional reduction of beam training overhead in crowded 802.11ad based networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 680–685.

[25] Muhammad Shoaib, Stephan Bosch, Ozlem D. Incel, Hans Scholten, and Paul J.M. Havinga. 2014. Fusion of Smartphone Motion Sensors for Physical Activity Recognition. *Sensors* 14, 6 (2014), 10146–10176.

[26] SIVERSIMA. 2018. EVK06002 Development Kit. (2018). https://www.siversima.com/product/evk-600200

[27] SIVERSIMA. 2019. EVK02001 28 GHz Development Kit. (2019). https://www.siversima.com/product/evk-02001-00/

[28] P. F. M. Smulders. 2009. Statistical Characterization of 60-GHz Indoor Radio Channels. *IEEE Transactions on Antennas and Propagation* 57, 10 (2009), 2820–2829.

[29] Daniel Steinmetzer, Daniel Wegemer, Matthias Schulz, Joerg Widmer, and Matthias Hollick. 2017. Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11ad Devices. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '17)*. Association for Computing Machinery, New York, NY, USA, 414–425. https://doi.org/10.1145/3143361.3143384

[30] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2017. WiFi-Assisted 60 GHz Wireless Networks. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17)*. Association for Computing Machinery, New York, NY, USA, 28–41. https://doi.org/10.1145/3117811.3117817

[31] Vadatech. 2018. AMC599 board. (2018). https://www.vadatech.com/media/AMC599_AMC599_Datasheet.pdf

[32] Vadatech. 2018. AMC726 board. (2018). https://www.vadatech.com/media/AMC726_AMC726_Datasheet.pdf

[33] IEEE 802.11 working group. 2012. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band. *IEEE Standard 802.11ad* (2012).

[34] IEEE 802.11 working group. 2019. IEEE Draft 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: Enhanced Throughput for Operation in License-Exempt Bands Above 45 GHz. *IEEE P802.11ay/D3.0* (2019).

[35] Jialiang Zhang, Xinyu Zhang, Pushkar Kulkarni, and Parameswaran Ramanathan. 2016. OpenMili: A 60 GHz Software Radio Platform with a Reconfigurable Phased-array Antenna. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking (MobiCom '16)*. ACM, New York, NY, USA, 162–175. https://doi.org/10.1145/2973750.2973760

[36] R. Zhao, T. Woodford, T. Wei, Q. Kun, and X. Zhang. 2020. M-Cube: A Millimeter-Wave Massive MIMO Software Radio. In *ACM MobiCom*.

[37] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y. Zhao, and Haitao Zheng. 2014. Demystifying 60GHz Outdoor Picocells. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom '14)*. Association for Computing Machinery, New York, NY, USA, 5–16. https://doi.org/10.1145/2639108.2639121