# STAMINA: Implementation and Evaluation of Software-Defined Millimeter Wave Initial Access

Joao F. Santos*, Efat Fathalla†, Aloizio P. DaSilva*, Luiz A. DaSilva*, and Jacek Kibilda*
*Commonwealth Cyber Initiative, Virginia Tech, USA, e-mail: {joaosantos, aloiziops, ldasilva, jkibilda}@vt.edu
†Old Dominion University, USA, e-mail: efath002@odu.edu

***Abstract*—In this paper, we present a framework for experimentation in next-generation Initial Access (IA) procedures for Millimeter Wave (mmWave) and Terahertz (THz) communications called SofTwAre-defined Mmwave INitial Access (STAMINA). The IA procedure is one of the essential components for communication systems in high frequencies, enabling directional transmitters and receivers to acquire each other's relative orientation before data transmission. While effective in establishing communication, the existing IA procedure standardized by 3GPP consumes a significant amount of radio resources. Many research efforts have proposed enhancements over the current-generation IA procedure, e.g., leveraging non-uniform beam sweep sequences or adaptive codebooks. However, no existing experimental mmWave platforms support modifications in their standard-compliant IA procedures, preventing their utilization for conducting experimental research on next-generation IA procedures. Our software-defined mmWave framework addresses this gap by combining the flexibility of Software-defined Radios (SDRs) with the directionality of mmWave front-ends to perform customizable IA procedures. We demonstrate STAMINA's ability to control mmWave front-ends correctly, its increased performance over traditional static experiments, and its flexibility to customize the IA parameters to achieve different objectives. Our results show that STAMINA provides experimenters with a flexible platform for performing experiments on next-generation IA procedures.**

***Index Terms*—Software-defined radio, Millimeter Wave, Initial Access, Next-generation Networks, Programmability**

## I. Introduction

The demand for services with lower latency and higher throughput, e.g., industrial automation, Unmanned Aerial Vehicle (UAV), and Augmented/Virtual Reality (AR/VR), pushes Radio Access Technologies (RATs) to explore higher frequencies, where the available bandwidth is vast, and the opportunities for innovation of next-generation access networks are many [1], [2]. However, as we go up to the Millimeter Wave (mmWave) range and above, with the expected inclusion of Terahertz (THz) communications in 6G and beyond [3], the free-space path loss becomes 1–3 orders of magnitude larger than current-generation mobile networks operating in sub-6 GHz spectrum [1], [4], [5]. To compensate for the reduced link budget, mmWave and THz communication systems leverage beamforming over large antenna arrays to generate narrow, high-powered directional *beams* able to traverse long unobstructed distances [1], [6]. On the one hand, the spatially-confined nature of the beams results in efficient spatial reuse of spectrum [7], [8]. On the other hand, such spatial signal separation imposes a new set of challenges for mmWave and THz communication systems, as transmitter and receiver must locate and point their beams toward one another to establish communication, and the link performance depends on maintaining a fine beam alignment [5], [9]–[11].

The 5G protocol includes an Initial Access (IA) procedure, where transmitter and receiver mmWave radios sequentially sweep over all possible pairs of their beams, spanning predefined angular directions during a fixed time interval. The transmitter sends Synchronization Signal Blocks (SSBs) containing known synchronization signals on each beam, and, at the end of the sweep, the receiver determines the best beam pair based on the highest Reference Signal Received Power (RSRP) [9]–[11] (detailed further in Section II). The IA procedure consumes 20-50% of airtime [12] depending on the spectrum band and antenna configuration used, which lowers the radio resources available for data transmission. In addition, the IA procedure standardized by the 3GPP is a static operation that cannot be customized or optimized based on the context or scenario [13]. Examples of customization include: (*i*) sweeping a subset of the available beams to reduce the overhead at the cost of accuracy [12]; (*ii*) using non-uniform beam sweep sequences to prioritize specific directions [14] (useful to serve fixed users); or (*iii*) employing adaptive codebooks and non-uniform beam durations to leverage Reflective Intelligent Surface (RIS) for expanding cell coverage or mitigating blockages [15].

Current efforts towards 6G networks advocate for flexible, open management frameworks with AI-native control loops, exposing APIs for ML-based solutions to optimize different aspects of network functionality, from mobility down to the PHY and MAC layers [16]–[19]. For example, the O-RAN Alliance is exploring AI to optimize the placement of SSBs in the 5G frame structure, and coordinate multiple mmWave radios with distinct capabilities for meeting different Key Performance Indicators (KPIs), e.g., the IA accuracy and decision delay [18]. An ML-driven next-generation IA able to optimize fundamental parameters related to IA on the fly, e.g., beam sweep sequences, beam duration, and payload duration, can improve efficiency, reduce costs, and expand the use cases envisioned for 6G [13]–[16]. However, to the best of the authors' knowledge, no existing open mmWave platforms expose APIs for customizing their standard-compliant IA procedures, hindering further experimental research from evaluating trade-offs related to IA parameters or collecting datasets for training ML-based solutions.

There are a few examples of research efforts that explored bridging (*i*) flexible and inexpensive Software-defined Ra-

dio (SDR) platforms that operate in the sub-6 GHz band, with (*ii*) mmWave front-ends that up/downconvert signals to/from the mmWave band, creating programmable mmWave platforms for rapid prototyping and testing in over-the-air (OTA) scenarios [2], [6], [8]. However, these works mainly focus on leveraging SDRs to control basic functionality of the mmWave front-ends, e.g., gain, transmission mode, and beam index, for experimentally characterizing their irradiation patterns [2], beamwidths [8], and channel reciprocity [6]. The lack of customizable IA functionality, including timing beam sweeps, collecting performance metrics, implementing decision algorithms, and selecting the best beam for data transmission, prevents using the existing solutions to conduct both link- and system-level experiments on the IA procedure.

In this work, we address the lack of a flexible mmWave platform to experimentally evaluate performance trade-offs in parameters associated with the IA procedure. We have developed the SofTwAre-defined Mmwave INitial Access (STAMINA), a software-defined mmWave framework based on GNU Radio [20] that enables experimentation on next-generation IA procedures for mmWave and THz communications. STAMINA allows experimenters to customize parameters associated with the IA procedure on the fly, e.g., beam sweep sequences, beam duration, and IA interval, as well as collect datasets with different KPIs to evaluate the impact of their changes, which can be used to train ML-based solutions to optimize parameters of the IA procedure. To the best of our knowledge, this is the first work to (*i*) abstract the IA procedure into a control loop that facilitates the analysis of IA trade-offs and design choices; and (*ii*) present an open-source solution to expose control of mmWave front-ends through General Purpose Input/Output (GPIO) interfaces in GNU Radio. For further information and download our software package, the reader can refer to STAMINA's open-source repository (https://github.com/CCI-NextG-Testbed/gr_stamina).

The remainder of this paper is organized as follows. In Section II, we detail the standard IA procedure and model it into a control loop with discrete components. In Section III, we introduce our flexible software-defined mmWave framework and describe its architecture. In Section IV, we experimentally evaluate STAMINA and validate its ability to customize the IA procedure on the fly. Finally, in Section V, we summarize our conclusions and discuss potential directions for future work.

## II. mmWave Initial Access Control Loop

The IA procedure serves to establish an initial connection between directional transmitters and receivers unaware of each other's relative orientation before data transmission [9]–[11]. We can decompose the IA procedure into six phases, as shown in Fig. 1. For simplicity, we assume transmitter and receiver possess directional antennas with full beam reciprocity for transmission and reception, allowing the reuse of the best transmit beam for reception and vice versa [11].

First, the transmitter periodically broadcasts known synchronization signals, each of which is mapped onto a different beam of the transmitter codebook and iteratively transmitted in all angular directions. During the transmission of each synchronization signal, the receiver scans the different beams of the receiver codebook. It conducts measurements to detect and decode the signal with the highest received power, indicating the receiver beam that corresponds to the best direction for reception. Next, the receiver randomly selects one random access preamble and transmits it using the beam with the best direction for reception [11]. During the preamble transmission phase, the transmitter scans the different beams of the transmitter codebook. It conducts measurements to detect random access preambles and the beam where the preambles obtained the highest received power, indicating the best direction for transmission [9]. Then, the transmitter sends an acknowledgment to the receiver using the beam with the best direction for transmission, containing the index of the detected random access preamble to confirm a successful IA procedure. Finally, the link is established with the transmitter and receiver aware of the best beam pair that corresponds to their relative orientation, and they transmit data [10]. The IA procedure is periodically triggered to fine-tune the beam alignment, anticipate user mobility, and recover from blockages [9].

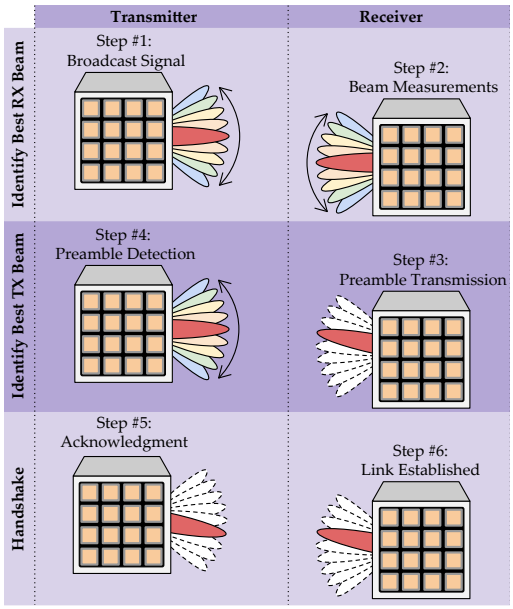



Fig. 1: Diagram of the IA procedure, enabling radios to identify the best beam pair according to their relative orientation.

The 5G protocol uses SSBs as its IA synchronization signal, containing the Primary Synchronization Signal (PSS), Secondary Synchronization Signal (SSS), and Physical Broadcast CHannel (PBCH). The decoded SSB also provides the receiver with frame timing, cyclic prefix length, among other parameters for data transmission [9]. Also, the 5G protocol uses the RSRP metric to decide the best beam pairs, i.e., selecting beams based on the highest received power of their reference symbols [9].

From the viewpoint that the IA procedure periodically iterates over different beams on the transmitter and receiver radios and selects the best beam pair for data transmission,

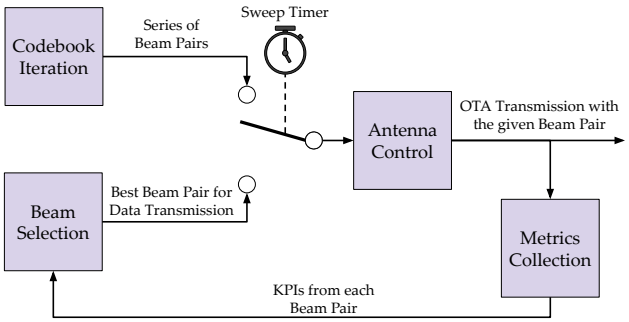



Fig. 2: Mapping of the IA procedure as a control loop.

we can simplify the IA procedure and abstract it as a control loop composed of four functional components, as illustrated in Fig 2. Without loss of generality, we assume the transmitter and receiver codebooks are defined prior to the IA procedure. The behavior of the control loop changes over time, performing different actions during beam sweep and data transmission. During beam sweep, a codebook iteration algorithm decides how to iterate over the available beams in the transmitter and receiver codebooks, and how to combine them to generate a series of beam pairs pointing in different directions. For each selected beam pair, an antenna control component interacts with the mmWave equipment to perform OTA transmissions using the given transmit and/or receiver beams. While the OTA transmissions occur during beam sweep, a metrics collection component captures KPIs from each beam pair. A timer expires after the transmission of all beam pairs, marking the end of the beam sweep. At this point, a beam selection component analyzes the collected KPIs and decides the best beam for data transmission. By breaking down the IA procedure into a control loop with discrete functional components, we can easily analyze how specific aspects of the IA procedure affect the overall system performance, e.g., using different codebook iteration algorithms, KPIs, or criteria for selecting the best beam pair, and assess improvements over the current-generation IA procedure.

## III. SoftWare-defined MmWave INitial Access

To enable experimental research on next-generation IA procedures for 6G networks, we developed STAMINA, a software-defined mmWave framework that serves as a platform for experimentation and data collection on IA. STAMINA provides programmable mmWave platforms with a complete and flexible IA control loop, including (*i*) controlling mmWave front-ends to sweep over arbitrary beam sweep sequences with custom durations; (*ii*) collecting different KPIs and reporting them with arbitrary cadence; (*iii*) implementing different decision algorithms to select the best beam pair for data transmission; and (*iv*) transmitting payload for an arbitrary duration before restarting the IA procedure. In the remainder of this section, we detail how we leveraged GNU Radio to implement STAMINA, and developed the different components of the IA control loop.

### A. Implementing STAMINA using GNU Radio

To simplify the development of STAMINA and facilitate its adoption by the research community, we implemented

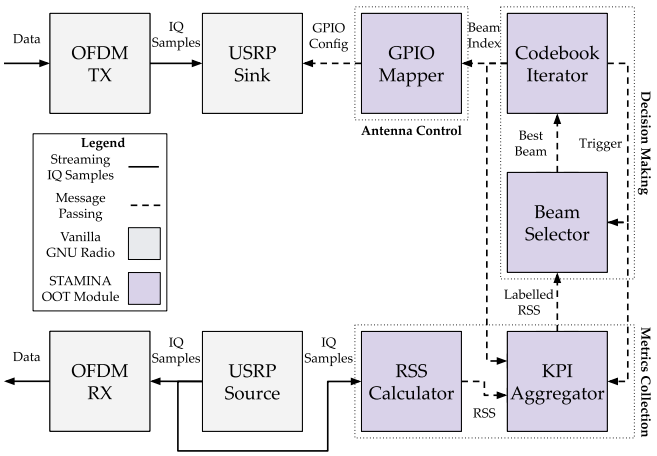



Fig. 3: STAMINA's architecture in GNU Radio: the complete and flexible IA control loop comprises a combination of our custom OOT blocks (purple) and vanilla blocks (gray).

STAMINA using GNU Radio, a widely known Software Development Kit (SDK) for experimenting and prototyping on SDRs [20]. We implemented STAMINA as a GNU Radio Out-of-Tree (OOT) module composed of control and signal processing blocks that serve different functions of the IA control loop, as shown in Fig. 3. These blocks can be used together to perform a complete, customizable IA procedure between a pair of mmWave front-ends attached to an SDR; or independently to evaluate the individual components of the IA control loop. Such modular design allows experimenters to easily customize or replace aspects of the IA procedure and focus only on the relevant implementation aspects of their experimental research. In addition, we leverage GNU Radio's reconfiguration capabilities to make STAMINA flexible, allowing experimenters to change the transmitter and receiver beam sweep sequences, the beam duration, the payload duration, and the measurement cadence at any point in time, and STAMINA will use the updated parameters in the following IA procedure.

Akin to the separation of the IA control loop into different components (discussed in Section II), we split STAMINA's blocks into three groups: antenna control, decision making, and metrics collection, as detailed in the following subsections.

### B. Antenna Control

This group of blocks is responsible for interfacing with the mmWave front-ends and controlling the physical hardware to sweep over a range of beams. It currently consists of a single block: the `GPIO Mapper`, detailed below.

*1) GPIO Mapper:* It abstracts the low-level interaction with the mmWave front-ends using the GPIO interface of the underlying SDR and exposes their high-level functionality in GNU Radio. This block serves two different purposes during startup and runtime. During startup, it initializes the mmWave front-ends and sets the transmission mode for each antenna, i.e., configuring it to act as either a transmitter or receiver. Then, during runtime, this block translates incoming messages with beam indices from the

`Codebook Iterator` into the necessary operations for configuring the desired beam on each array. These operations involve manipulating the SDR's GPIO banks to toggle or pulse certain pins to control the mmWave front-ends and beamforming vectors. The location and purpose of the GPIO pins and the required pulse duration are platform-dependent and vary according to different vendors and models. Thus, to enable the use of STAMINA by the broader research community, we parameterized the location and purpose of the GPIO pins using a configuration file that can be tailored according to the API specifications of different mmWave front-ends.

### C. Decision Making

This group of blocks is responsible for the intelligence in the control loop. Based on information from the metrics collection blocks, it decides the best beam pair and instructs the antenna control blocks to use this pair for data transmission. It consists of two blocks: the `Codebook Iterator` and the `Beam Selector`, detailed below.

*1) Codebook Iterator:* It decides how to iterate over the transmitter and receiver codebooks to generate beam pairs, timing how long the mmWave front-ends use each beam pair and how long to wait until the following IA procedure. Experimenters can select arbitrary beam sweep sequences for transmitter and receiver, with a custom beam duration for each beam pair and a payload duration for data transmission between IA procedures. This block begins IA procedures, using the GNU Radio message passing interface to trigger other relevant blocks, e.g., the `Beam Selector` and the `KPI Aggregator`, to start working (detailed later in this section). Then, it executes a codebook iteration algorithm; the default algorithm iterates over all combinations of transmitter and receiver beam sweep sequences to generate beam pairs. Next, it sends one beam pair at a time to the `GPIO Mapper`, waiting for the predefined beam duration before sending the next beam pair. After one beam sweep, i.e., iterating over all the chosen combinations of beam pairs, it triggers other blocks to halt (detailed later in this section). Finally, this block waits for the `Beam Selector` to provide the best beam pair for data transmission, which it forwards to the `GPIO Mapper` and waits for a given payload duration before starting a new IA procedure.

*2) Beam Selector:* It implements the decision mechanism for selecting the best beam for data transmission. During beam sweeps, this block stores incoming labeled KPIs according to their beam pair. At the end of an IA procedure, this block parses the collected KPIs, decides the best beam pair for data transmission until the start of the following IA procedure, and sends this information to the `Codebook Iterator`. In the current implementation of this block, it selects the beam pair with the highest Received Signal Strength (RSS) (detailed later in this section). However, since it holds all the measured KPIs captured during the last beam sweep, a different decision mechanism may be adopted, e.g., the highest average RSS per beam pair, an entirely different KPI (or a combination of KPIs), or an ML-based solution, to decide the best beam for data transmission.

### D. Metrics Collection

This group of blocks is responsible for collecting KPIs of different beam pairs. It consists of two blocks: the `RSS Calculator` and the `KPI Aggregator`, detailed below.

*1) RSS Calculator:* It continuously calculates the RSS of the received signal measured at the output of the receiver SDR. This block does so by computing a moving average of the received signal's squared magnitude (with a customizable averaging window). We decided to use the RSS as the main metric to showcase STAMINA's flexible IA control loop due to its RAT-agnostic nature, as it is independent of the signal bandwidth and modulation scheme. However, experimenters can replace this block with others that capture RAT-specific metrics, e.g., RSRP or Channel State Information (CSI), or include these new blocks for collecting additional KPIs during beam sweep.

*2) KPI Aggregator:* It aggregates KPI measurements from different sources, e.g., `RSS Calculator`, and labels them with the beam pair used by the mmWave front-ends. During beam sweeping, this block obtains the beam indices of the beam pair used for transmission and uses this information to label incoming KPIs, which ensures the correct attribution of KPIs to each beam pair. Then, it forwards the labeled measurements to the `Beam Selector` with an arbitrary cadence (detailed later in Section IV). At the end of an IA procedure, this block disregards incoming KPI measurements until the following IA procedure. User can also specify a minimum sensitivity for each input to discard measurements too low to be considered relevant. This block makes our STAMINA's metric collection flexible and future-proof, enabling experimenters to incorporate different metrics into their custom IA control loop by providing a simple method to label and feed KPIs to the `Beam Selector`.

STAMINA provides experimenters with a complete and flexible IA control loop. It enables the experimental evaluation of IA design trade-offs, e.g., assessing the effects of spending more time sweeping or transmitting data, and serves as a framework for exploring potential enhancements over current-generation IA procedures on the fly. For example, (*i*) testing sophisticated codebook iteration algorithms that sweep a subset of the available beams or prioritize specific directions [21]; (*ii*) leveraging different decisions algorithms more robust to outlier measurements [12]; or (*iii*) creating adaptive IA procedures, using ML-based solutions to customize or optimize IA parameters over time.

## IV. Experimental Proof-of-concept

In this section, we validate the use of STAMINA for controlling mmWave front-ends and performing flexible IA procedures. First, we detail our experimental setup and the equipment used in our evaluations. Then, we verify STAMINA's ability to interface with mmWave front-ends and perform beam sweeps. Next, we evaluate the benefits of using STAMINA to perform IA over static or manual beam configurations. Finally, we explore STAMINA's flexibility to customize IA procedures, showing design trade-offs according to different performance metrics.

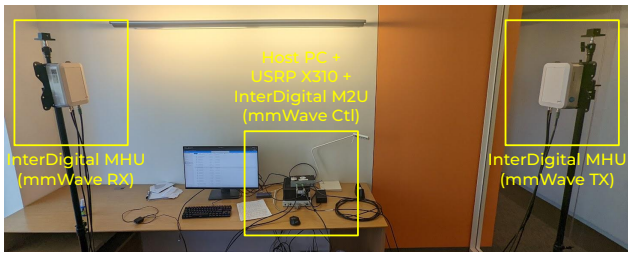



Fig. 4: Experimental setup we used to develop and evaluate STAMINA's proof-of-concept implementation.

### A. Experimental Setup

To develop and evaluate STAMINA, we leveraged the radio and computational resources available for remote experimentation in the Commonwealth Cyber Initiative (CCI) xG Testbed, in addition to a pair of mmWave front-ends and a mmWave controller provided by InterDigital, creating a programmable mmWave platform illustrated in Fig. 4, and detailed in Fig. 5. We used a host PC running GNU Radio and STAMINA's OOT module for baseband processing and controlling the mmWave equipment. We attached the host PC to a USRP X310 equipped with two UBX160 daughterboards that operate in the 10 MHz–6 GHz frequency range with a bandwidth of up to 160 MHz.

The mmWave equipment comprises two mmWave front-ends, the Mast Head Units (MHUs), and a mmWave controller, the MHU to USRP (M2U). The MHUs up/down convert signals with bandwidth up to 200 MHz from/to an Intermediate Frequency (IF) in the n46 band (5.3 GHz) to/from the n257 band (28 GHz). The MHUs possess an $8 \times 8$ phased array antenna with a gain of 23 dBi for transmitting and/or receiving, and contain two separate codebooks: a predefined codebook and a user-defined codebook. The predefined codebook comprises 63 calibrated beams arranged in a $9 \times 7$ grid, ranging from $\pm 45°$ in the azimuth and $\pm 35°$ in the elevation planes, with an angular resolution of $11.25°$ and $11.67°$, respectively. The user-defined codebook allows users to set custom antenna weight vectors offline before using the mmWave equipment. We used the predefined codebook in our experiments. The M2U provides the MHUs with power and synchronization (leveraging a $\pm 38$ ppb Oven-controlled Crystal Oscillator), and controls their transmission mode and current beam through a GPIO interface. We attached each MHU to the TX/RX port of each of our daughterboards, allowing us to change the transmission mode of both the mmWave front-ends and the SDR from software alone. In addition, we connected the GPIO port of the X310 to the M2U using a DB-15 cable. Finally, we placed the MHUs at 2.5 meters from each other and used OFDM frames with 1 MHz of bandwidth and 64 subcarriers for running the experiments in this initial evaluation.

### B. Controlling the mmWave Front-ends

In this analysis, we are interested in verifying STAMINA's ability to control the mmWave front-ends and perform beam sweeps. It is fundamental to validate STAMINA's ability to leverage the SDR's GPIO interface to correctly configure

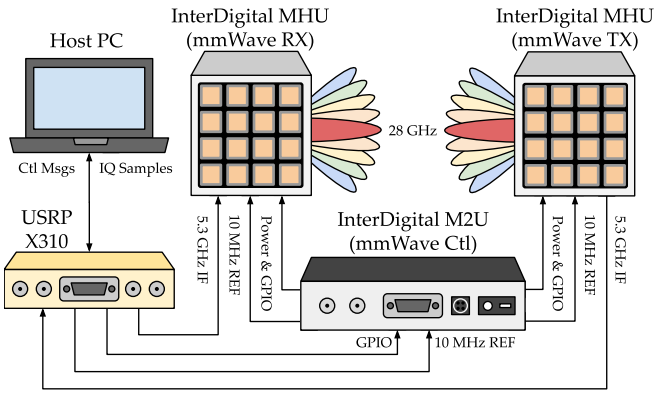



Fig. 5: System design detailing our experimental setup's hardware components and their connections.

the mmWave front-ends according to their API specification, setting their transmission mode, and changing beam pairs. To verify the correct configuration of the mmWave front-ends, we conducted three experiments, where we configured one of the MHUs to act as a transmitter and the other to act as a receiver, performed beam sweeping only on the azimuth plane, physically rotated the receiver MHU to different directions, and measured the average RSS for different beam pairs. Fig. 6 shows the results of our measurements. When the MHUs are in each other's boresight, we accurately obtained the highest average RSS of -5.65 dBm using the beam pair pointing directly at each other. Then, as we rotated the receiver counterclockwise ($-30°$) and clockwise ($+30°$) relative to the transmitter, we observed the highest average RSS shifted to the receiver beams in the $+33.75°$ and $-33.75°$ directions, accurately pointing towards the transmitter within the angular resolution of the mmWave front-ends, and obtained -8.51 dBm and -6.78 dBm, respectively. These results show STAMINA can accurately control the mmWave front-ends and perform beam sweeps, probing the environment to identify the beam pairs representing the best direction for data transmission.

### C. Probing the Environment to Find the Best Beams

In this analysis, we are interested in demonstrating the benefits of using STAMINA to perform IA procedures and find the best beam for data transmission, compared to performing experiments with a static beam configuration or a manual beam selection. For this purpose, we conducted three experiments, where we configured one MHU to act as a transmitter and the other MHU as a receiver, fixed the transmitter beam to its antenna's boresight, and captured measurements as we rotated the receiver in the azimuth plane using $11.25°$ increments. Fig. 7 shows the results of our measurements. First, we used a static boresight beam on the receiver, and observed a narrow region with high RSS (-5.65 dBm), decreasing by 11.52–14.71 dB as we rotated the receiver by $\pm 11.25°$, and dropping even further as the misalignment grows. Then, we manually selected the receiver beam toward the transmitter according to their relative orientation, known a priori. We observed a wider area with high RSS (-8.50 to -4.81 dBm) within a region of $\pm 45°$ of

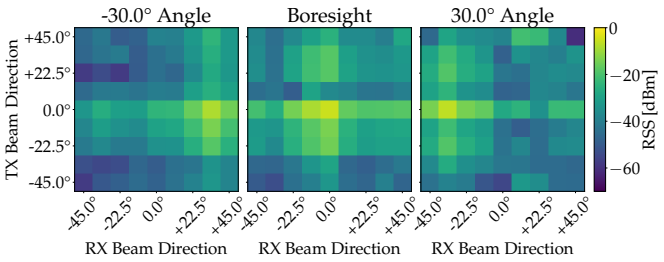



Fig. 6: Heatmaps showing the average RSS obtained from sweeping all combinations of beam pairs in the azimuth plane for a fixed transmitter, when the receiver is in the boresight of the transmitter's antenna and rotated $\pm 30°$ relative to the transmitter.

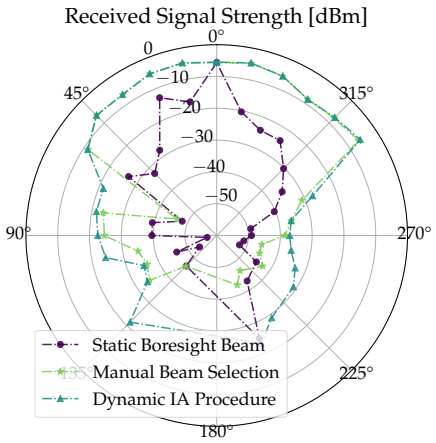



Fig. 7: Maximum RSS according to the receiver's relative orientation to a fixed transmitter: statically set to the boresight beam (purple), manually set to different beams (yellow), and dynamically adapted via IA procedures (blue).

relative orientation. Finally, we used STAMINA to perform an IA procedure on the receiver, and obtained similar results to the manual beam selection in the $\pm 45°$ region, but without requiring prior knowledge of the relative orientation between transmitter and receiver. This experiment also demonstrates how we can achieve better performance by probing the environment to leverage reflections, as we obtained RSS values 0.05–25.03 dB above the manual beam selection in regions without a direct line of sight transmission ($45°$–$315°$ of relative orientation). These results show the benefits of using STAMINA to perform experiments where (*i*) the radios' location and relative orientation are not known or set a priori; and (*ii*) we can dynamically probe the environment and leverage reflections to increase performance.

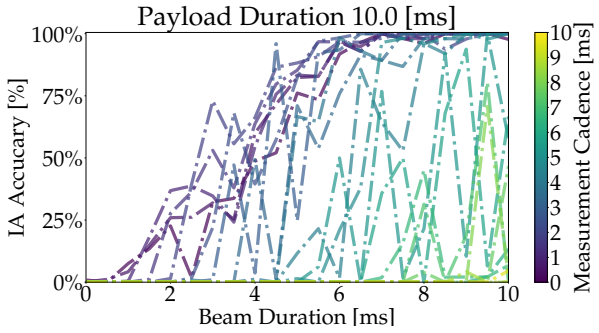
### *D. Exploring the IA Design Trade-offs*

In this analysis, we are interested in exploring STAMINA's flexibility to customize IA procedures and experimentally show IA design trade-offs. The parameters we mentioned earlier in Section III, e.g., the beam duration, the payload duration, and the measurement cadence affect the behavior of the IA procedure in different manners. To understand how these parameters affect the performance of IA procedures, we conducted a series of experiments where we configured one MHU to act as a transmitter and the other to act as a receiver, performed beam sweeping only on the azimuth plane, and captured performance metrics using different beam durations, payload durations, and measurement cadences. Fig. 8a shows how a longer beam duration increases the accuracy of the IA procedure, as it leads to the transmission of more OFDM frames per beam pair. However, this comes at the cost of consuming radio resources for data transmission, as shown in Fig. 8b, where the packet rate decreases with the increase in beam duration. We can also observe saturation in the IA accuracy at a beam duration of 6 ms, leaving enough radio resources to transmit four packets per second (note that we are using narrow 1 MHz OFDM frames with 64 subcarriers).

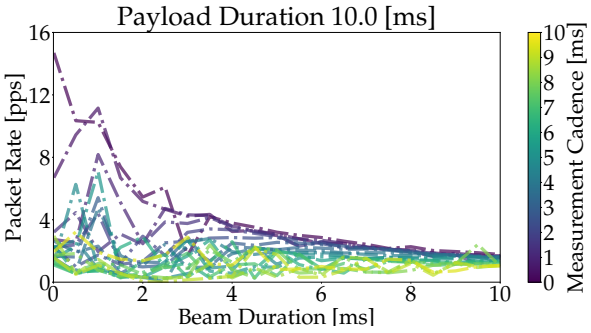
As we transmit multiple OFDM frames per beam pair, and multiple beam pairs per beam sweep, we amass a significant amount of data before deciding on the best beam pair. To decrease this computational footprint, we introduced a measurement cadence, i.e., a rate at which we label measurements and feed them to the `Beam Selector`. In the previous evaluations, lower measurement cadences led to better performance, as we obtained more timely information about beam sweeps. However, this comes at the cost of accumulating too many measurements and taking longer to analyze the collected KPIs to decide the best beam pair, as shown in Fig. 8c. We can observe that the decision delay can become non-negligible, as it saturates around 7 ms on our setup. The combination of beam duration, payload duration, and the decision delay dictate the IA procedure's total duration, i.e., the delay in updating beam pairs to transmit data, as shown in Fig 8d. The update delay impacts the communication latency and the performance of mobile users. We can observe how the payload duration affects the update delay until a beam duration of 1 ms; from this point forward, the beam duration becomes the most significant component. These results demonstrate STAMINA's flexibility to create custom IA procedures, allowing experimenters to (*i*) evaluate the performance impact of different IA parameters and (*ii*) create IA procedures tailored to the performance requirement of their own applications.

## V. Conclusions

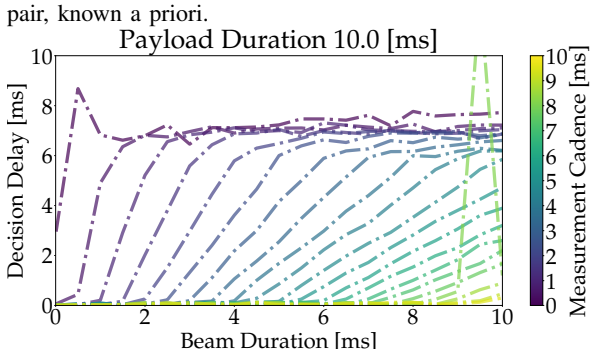
In this paper, we presented STAMINA, a software-defined mmWave framework that enables the experimental evaluation of performance trade-offs in IA parameters towards next-generation IA procedures. First, we examined the current-generation IA procedure and proposed its abstraction as a control loop composed of discrete functional components. Then, we detailed STAMINA's implementation using GNU Radio, and how we developed the different components of the IA control loop to perform flexible IA procedures using a combination of SDRs and mmWave front-ends. We evaluated STAMINA's ability to control mmWave front-ends correctly, its performance advantages over traditional static experiments, and its flexibility for tailoring IA parameters to achieve different objectives. In future works, we are investigating ML-based solutions to optimize IA parameters, and exposing the control over STAMINA to the Near Real-Time



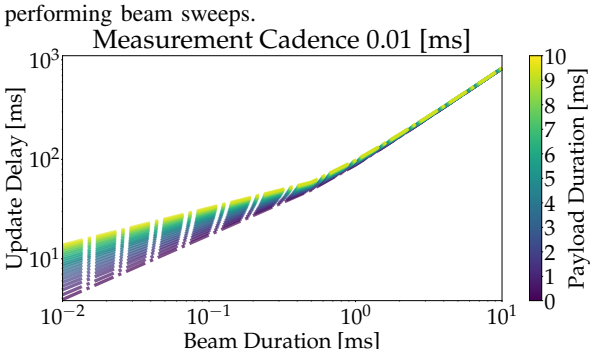

(a) Accuracy of the IA procedure to select the correct beam pair, known a priori.




(b) Average packet rate obtained when transmitting data after performing beam sweeps.




(c) Average delay for the `Beam Selector` to decide the best beam for data transmission.




(d) Total delay to obtain updated beam pairs, resulting from the IA parameters and decision delay.

Fig. 8: Performance metrics of the IA procedure according to beam durations, payload durations, and measurement cadences.

RAN Intelligent Controller (RIC) in the O-RAN ecosystem through a customized E2 interface in GNU Radio.

## ACKNOWLEDGMENTS

The research leading to this paper received support from the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation, and workforce development. For more information about CCI, visit: www.cyberinitiative.org. In addition, we would like to thank InterDigital for providing us with their mmWave equipment.

## REFERENCES

[1] K. Zheng *et al.*, "Software-defined Radios to Accelerate mmWave Wireless Innovation," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–4.

[2] T. Chen *et al.*, "Programmable and Open-Access Millimeter-Wave Radios in the PAWR COSMOS Testbed," in *ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH)*, 2022, pp. 1–8.

[3] K. Rikkinen *et al.*, "THz Radio Communication: Link Budget Analysis Toward 6G," *IEEE Communications Magazine (ComMag)*, vol. 58, no. 11, pp. 22–27, 2020.

[4] M. Polese *et al.*, "MillimeTera: Toward a Large-scale Open-source mmWave and Terahertz Experimental Testbed," in *ACM Workshop on Millimeter-wave Networks and Sensing Systems (mmNets)*, 2019, pp. 27–32.

[5] J. Kibiłda *et al.*, "Indoor Millimeter-wave Systems: Design and Performance Evaluation," *Proceedings of the IEEE (PIEEE)*, vol. 108, no. 6, pp. 923–944, 2020.

[6] A. Gaber *et al.*, "USRP-based mmWave Prototyping Architecture with Real-Time RF Control," in *IEEE International Microwave Symposium (IMS)*, 2021, pp. 693–696.

[7] Q. Chen *et al.*, "Spatial Reuse Strategy in mmWave WPANs with Directional Antennas," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 5392–5397.

[8] M. Danneberg *et al.*, "USRP-based Platform for 26/28 GHz mmWave Experimentation," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2020, pp. 1–6.

[9] M. Giordani *et al.*, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 21, no. 1, pp. 173–196, 2018.

[10] I. Aykin and M. Krunz, "Efficient Beam Sweeping Algorithms and Initial Access Protocols for Millimeter-wave Networks," *IEEE Transactions on Wireless Communications (TWC)*, vol. 19, no. 4, pp. 2504–2514, 2020.

[11] C. N. Barati *et al.*, "Initial Access in Millimeter Wave Cellular Systems," *IEEE Transactions on Wireless Communications (TWC)*, vol. 15, no. 12, pp. 7926–7940, 2016.

[12] T. S. Cousik *et al.*, "Deep Learning for Fast and Reliable Initial Access in AI-Driven 6G mm Wave Networks," *IEEE Transactions on Network Science and Engineering (TNSE)*, 2022.

[13] J. Liu *et al.*, "Initial Access, Mobility, and User-centric Multi-beam Operation in 5G New Radio," *IEEE Communications Magazine (ComMag)*, vol. 56, no. 3, pp. 35–41, 2018.

[14] A. Mazin *et al.*, "Accelerating Beam Sweeping in mmWave Standalone 5G New Radios Using Recurrent Neural Networks," in *IEEE Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–4.

[15] P. Wang *et al.*, "Beam Training and Alignment for RIS-assisted Millimeter Wave Systems: State of the Art and Beyond," *IEEE Wireless Communications (WCM)*, vol. 29, no. 6, pp. 64–71, 2022.

[16] ATIS. (2022, Feb.) NextG Alliance Report: Roadmap to 6G. NextG Alliance. Accessed on Oct. 23, 2022. [Online]. Available: https://www.nextgalliance.org/wp-content/uploads/2022/02/NextGA-Roadmap.pdf

[17] J. F. Santos *et al.*, "Breaking Down Network Slicing: Hierarchical Orchestration of End-to-End Networks," *IEEE Communications Magazine (ComMag)*, vol. 58, no. 10, pp. 16–22, 2020.

[18] O-RAN Alliance, "O-RAN Working Group 1: Massive MIMO Use Cases Technical Report," O-RAN Alliance, Tech. Rep., Jul. 2022.

[19] J. Hoydis *et al.*, "Toward a 6G AI-native air interface," *IEEE Communications Magazine (ComMag)*, vol. 59, no. 5, pp. 76–81, 2021.

[20] E. Blossom, "GNU Radio: Tools for Exploring the Radio Frequency Spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.

[21] F. Morandi *et al.*, "A Probabilistic Codebook Technique for Fast Initial Access in 6G Vehicle-to-vehicle Communications," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.