
# *LiBRA*: Learning-Based Link Adaptation Leveraging PHY Layer Information in 60 GHz WLANs

Shivang Aggarwal, Urjit Satish Sardesai, Viral Sinha, Deen Dayal Mohan, Moinak Ghoshal, Dimitrios Koutsonikolas
University at Buffalo, The State University of New York, NY, USA
{shivanga,urjitsat,viralsin,dmohan,moinakgh,dimitrio}@buffalo.edu

## ABSTRACT

We conduct one of the first extensive experimental studies of the two main link adaptation mechanisms in 60 GHz WLANs, namely *rate adaptation* and *beam adaptation*. We first show, using a variety of commodity 60 GHz devices, that simple heuristics, used by these devices to determine which the two mechanisms should be triggered, can lead to wrong decisions even in seemingly simple scenarios. We then explore for first time the feasibility of leveraging PHY layer information and ML to guide link adaptation, using a large dataset collected with a 60 GHz software-define radio testbed in a variety of indoor environments and scenarios. Finally, we design *LiBRA*, a practical, standard-compliant link adaptation framework that leverages ML and PHY layer information to determine when to trigger link adaptation and which adaptation mechanism to use. *LiBRA* strikes a balance between throughput and link recovery delay, performing close to an oracle solution, and outperforming significantly two simple heuristics used by off-the-shelf devices.

## CCS CONCEPTS

• **Networks** → **Link-layer protocols**; **Network performance analysis**; **Wireless local area networks**.

**ACM Reference Format:**
Shivang Aggarwal, Urjit Satish Sardesai, Viral Sinha, Deen Dayal Mohan, Moinak Ghoshal, Dimitrios Koutsonikolas. 2020. *LiBRA*: Learning-Based Link Adaptation Leveraging PHY Layer Information in 60 GHz WLANs. In *The 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20), December 1–4, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3386367.3431319

## 1 INTRODUCTION

Rate adaptation (RA) – dynamically adjusting the modulation and coding scheme (MCS) based on the channel quality – has been traditionally considered as the main link adaptation mechanism in 802.11-based WLANs. Since the introduction of the first RA algorithm for 802.11b [36], all 802.11 standards mandate the support for multiple rates at the PHY layer.

Millimeter-wave (mmWave) wireless is fast emerging as the prime candidate technology for providing wireless multi-Gbps data rates. For example, the IEEE 802.11ad and the upcoming 802.11ay WLAN standards [32, 33] utilize the 14 GHz of unlicensed spectrum around 60 GHz and provide data rates of up to 6.7 Gbps and 30 Gbps, respectively. To cope with the high propagation loss in the mmWave frequency bands, mmWave transceivers establish highly directional links. However, high directionality introduces new challenges – vulnerability to blockage and beam misalignment due to mobility. While RA continues to play a critical role in the performance of 60 GHz WLANs, it cannot always address these new challenges alone. Hence, 60 GHz radios are typically equipped with electronically steerable phased antenna arrays, and employ a second link adaptation mechanism, beam adaptation (BA), also referred to as *beamforming* or *beam searching*, to maintain Tx-Rx beam alignment. However, the standards do not specify *when each of the two adaptation mechanisms, RA or BA, should be used or in what order*, and 60 GHz chipset vendors resort to simple heuristics to select the right mechanism.

In this paper, we fill this gap by conducting one of the first extensive experimental studies of the two link adaptation mechanisms – RA and BA – in 60 GHz WLANs. Our primary goal is to answer the question: *Can we provide guidelines on which of the two adaptation mechanisms should be triggered under a variety of scenarios involving different link impairments?* It turns out that this seemingly simple question is very hard to answer in practice. We conduct our study in 5 steps:

- Using a variety of commercial off-the-shelf (COTS) 802.11ad devices, we evaluate the effectiveness of the heuristics such devices use to determine when to trigger each of the two adaptation mechanisms. We find that *these heuristics often make wrong decisions, which in turn lead to reduced performance, even in seemingly simple scenarios.*
- *We explore for first time the feasibility of utilizing PHY layer information to guide link adaptation in 60 GHz WLANs* using a large dataset[1] collected with the X60 testbed [50] in a variety of indoor environments and scenarios. We investigate the effectiveness of a number of PHY layer metrics in predicting which of the two mechanisms should be triggered at a given scenario. While some metrics turn out to be more useful than others, our study reveals that no metric works in all scenarios, suggesting that a combination of metrics is required.
- *We explore for first time ML-based link adaptation approaches.* We test different popular classical and deep-learning methods using our dataset and 5-fold cross-validation and show that simple models based on random forests can predict the right action with 98% accuracy. We further evaluate the models on a new dataset

[1]The dataset is publicly available at http://bit.ly/60ghz-link-adaptation.


*CoNEXT '20, December 1–4, 2020, Barcelona, Spain*

ACM ISBN 978-1-4503-7948-9/20/12…$15.00
https://doi.org/10.1145/3386367.3431319

collected from two different buildings showing that they retain satisfactory accuracy.
- We design *LiBRA*, a *practical, standard-compliant, learning-based* link adaptation framework that leverages PHY layer information to determine (i) when to trigger link adaptation and (ii) which of the two adaptation mechanisms to trigger, and works with a variety of RA and BA algorithms. *LiBRA* strikes a balance between two performance metrics – throughput and link recovery delay.
- We evaluate *LiBRA* using extensive trace-based simulations with different sets of realistic PHY and MAC layer parameters. Our results show that *LiBRA* performs closely to an oracle that always selects the right adaptation mechanism depending on the performance metric one wants to optimize and significantly outperforms two simple heuristics, in a variety of indoor scenarios.

## 2 BACKGROUND AND RELATED WORK

**Rate Adaptation.** The goal of RA is to select a PHY data rate, expressed as a combination of a modulation and a coding scheme (MCS), that matches the Rx channel quality. Since the Rx channel quality is not known on the Tx side without explicit feedback, the majority of RA algorithms for legacy WiFi [2, 8, 18–22, 29, 35, 37–39, 42, 45, 47, 51, 58, 62, 62, 63, 67] estimate the channel quality using link layer statistics and employ simple heuristics based on these statistics. The use of SNR and other PHY layer metrics has also been proposed in the literature [19, 21, 25, 35, 51, 58, 67] but has not been used by WiFi chipset vendors.

The 802.11ad standard defines 12 MCSs for data frame transmission for the single-carrier (SC) PHY that is used by all COTS devices [1, 4–7], yielding data rates from 385-4620 Mbps. In contrast to legacy WiFi, RA has not been extensively studied in the context of 60 GHz WLANs. A few works [24, 27, 56, 71, 72] have argued that 60 GHz links are more stable than legacy WiFi links, and suggested the use of simple SNR-based RA algorithms via a direct SNR-MCS mapping. However, in our recent work, we showed experimentally that MCS is only weakly correlated with SNR in 60 GHz WLANs [49, 50] and SNR-based RA performs poorly in real-world indoor settings [9]. COTS 60 GHz devices, on the other hand, use heuristics similar to those used by legacy WiFi devices, e.g., they lower the MCS upon frame loss [14].

**Beam Adaptation.** The goal of BA is to find the Tx-Rx beam (sector) pair that maximizes the SNR. A naive approach is to test all possible pairs, but the overhead of this approach ($O(N^2)$, where $N$ is the number of available beams) can be prohibitive (up to a few seconds [56]) in the case of a large number of available beams. A number of recent works have proposed algorithms to reduce the overhead [11, 24, 28, 31, 43, 54, 57, 70]. The 802.11ad standard takes a different approach having each side train their Tx and Rx beams separately [32]. First, Tx beam training takes place where each side performs a sector level sweep (SLS), while the other side receives in quasi-omni mode. Rx beam training follows, where each side transmits in quasi-omni mode while the other side performs an SLS in receive mode. This approach reduces the complexity from $O(N^2)$ down to $O(N)$. To further simplify the process, COTS devices only perform Tx beam training and always receive in quasi-omni mode, further cutting down the overhead by half.

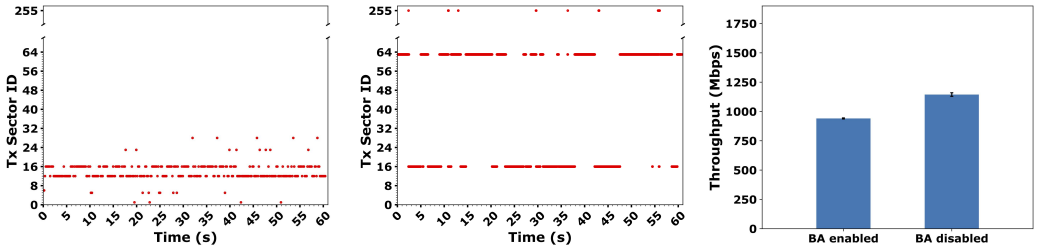
Surprisingly, the 802.11ad/ay standards do not specify when each of the two mechanisms should be triggered, and the problem has been largely overlooked by the research community. COTS devices trigger RA in the case of a missing ACK and only resort to BA if a working MCS cannot be found [49]. In [14], it is pointed out that this approach may often be suboptimal because of the overhead of trying all possible MCSs. Also, even if RA eventually finds a working MCS, that MCS may still not result in optimal throughput, since a different beam may support a higher MCS. Hence, another approach is proposed in [14] that first performs BA in the event of link degradation and then RA. In practice, *none of the two approaches can always guarantee optimal performance*. In certain cases (e.g., a client turning away from the AP), BA is required to restore the link, while in other cases (e.g., a client moving backwards facing the AP), the optimal beam may remain the same and triggering BA may result in unnecessary overhead. In our recent work [9], we further showed that often none of the two mechanisms is sufficient alone and both are required for optimal performance. However, unlike in this work, we did not explore in [9] the problem of selecting which of the two mechanisms to trigger first. The only other work that has looked into the problem is [24], which proposes beam sounding – a non-standard-compliant approach, based on a new control frame. In contrast, in this work, we are proposing a *standard-compliant* learning-based approach leveraging PHY layer information.

**ML in mmWave systems.** Recent works have used ML in mmWave cellular systems for blockage prediction [10, 12, 13, 15, 59], distinguishing blockage from mobility [65], or channel classification [40], to guide RA [40] or BA [10, 13, 15, 59] *separately*, and to trigger handoffs [12]. In our recent work [9], we also explored for first time the use of ML to guide RA in 60 GHz WLANs. We showed that this approach works better than SNR-based RA but it is environment-dependent and requires online training. Different from all these works, *LiBRA leverages ML to select the right adaptation mechanism in a variety of scenarios involving blockage, mobility, or interference, and works with both classical and learning-based RA and BA algorithms*. Also, in contrast to the works in [10, 12, 13, 15, 40, 59, 65] that rely on simulations, this work performs an experimental exploration of BA and RA using both COTS 60 GHz hardware and a software defined radio (SDR) testbed.

## 3 MOTIVATION

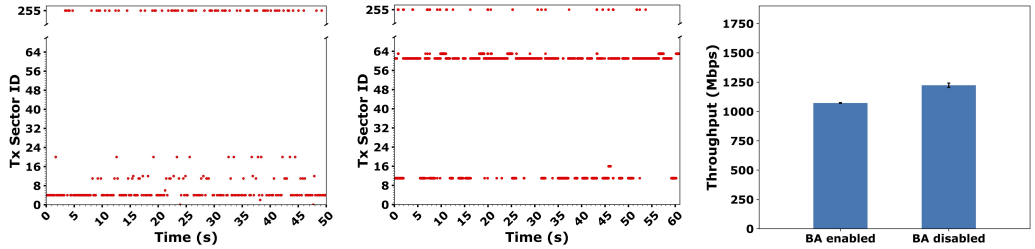
In this section, we use controlled experiments to study BA and RA in COTS devices equipped with 802.11ad radios. Our setup consists of a TP-Link Talon AD7200 [4] WiFi router and two different client devices – an Acer Travelmate P446-M laptop [1] and an ASUS ROG Phone [6]. All three devices first perform RA if no Block ACK is received after an Aggregated Frame (AMPDU) transmission, and trigger BA if no working MCS is found. We flashed the Talon router with the modified LEDE firmware from [53] that supports disabling the BA process and manual setting of a sector. We cannot control BA on the client devices and RA on any device. We experimented with uplink and downlink TCP iperf sessions. We found that the laptop behaves very similarly to the AP, since both devices have the same chipset and phased array. Hence, we only show results for the AP-laptop links (downlink) and phone-AP links (uplink).

**Static settings.** This is the simplest scenario, where the client remains static facing the AP, without any blockage or interference.



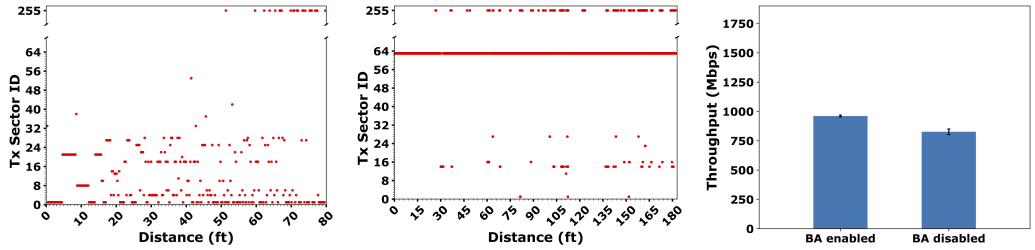

(a) Tx sector selection on the phone. (b) Tx sector selection on the AP. (c) Throughput with/without BA.

**Figure 1: Tx Sector variations in static scenarios.**




(a) Tx sector selection on the phone. (b) Tx sector selection on the AP. (c) Throughput with/without BA.

**Figure 2: Tx sector variations in blockage scenarios.**




(a) Tx sector selection on the phone. (b) Tx sector selection on the AP. (c) Throughput with/without BA.

**Figure 3: Tx sector variations in mobile scenarios.**

Hence, after an initial SLS, the Tx should lock on the best beam and never trigger BA, since any packet loss is due to transient drops in channel quality. We experimented in a corridor placing the client at different distances from the AP. Fig. 1a shows an example of a 60 s timeline of the Tx sector selection on the phone, when it is placed 30 ft away from the AP. We observe that, even in this simple scenario, the phone fails to make the right decision, triggering BA more than 100 times within a 60 s period and trying 6 different sectors. Fig. 1b shows a similar timeline on the AP for an AP-laptop downlink flow. Even though the sector selection on the AP is much more stable than on the phone, the AP also fails to lock on a single sector and triggers BA repeatedly, switching multiple times between two sectors over the 60 s period. Fig. 1c compares the throughput (averaged over 5 experiments) for the AP-laptop link when BA is enabled and when BA is disabled and the AP is locked on the best sector (manually discovered by sequentially trying all sectors). Disabling BA results in a 26% throughput improvement.

**Blockage.** We experiment with blockage scenarios in a lobby, where a TCP session starts with a human standing on the line-of-sight (LOS) between the AP and the client. Since the LOS is blocked, one would expect BA to be triggered only once at the beginning to discover a NLOS path via a reflection. Nonetheless, Figs. 2a, 2b show that the phone and the AP trigger BA multiple times and switch among a total of 4-5 sectors over a 50-60 s period, while in many cases they completely fail to lock on any sector (sector ID 255). The BA overhead results on average in 16% lower throughput for the AP-laptop session compared to the case of using the best static sector, as shown in Fig. 2c.

**Mobility.** We consider a simple mobility scenario where the client starts in front of the AP and moves away at walking speed while facing the AP, and a TCP iperf session runs for the duration of the motion. Since the client-AP orientation remains fixed for the duration of the motion, one would expect the Tx sector to remain fixed too. Nonetheless, Figs. 3a, 3b show that both the phone and the AP fail to lock on a single sector. However, in contrast to the static LOS and blockage cases, Fig. 3c shows that BA results in 15% higher throughput on average than using the best static sector. Hence, it is not always easy to decide whether BA should be disabled or not, even in LOS scenarios, as the imperfect beam patterns in COTS devices may result in an indirect path via a reflection to perform better than the direct path.

**Conclusion.** *Selecting the right link adaptation mechanism (BA vs. RA) in 60 GHz WLANs is very challenging in practice, even in seemingly simple scenarios. COTS devices often fail to make the right decision, yielding suboptimal performance.*

# 4 EXPERIMENTAL SETUP

## 4.1 X60 Testbed

COTS 802.11ad-compliant devices do not allow us to disable RA and to access the PHY layer. Hence, in the remainder of the paper we use the SDR-based X60 [50] testbed. X60 is the only 60 GHz testbed that combines fully programmable PHY and MAC layers, multi-Gbps data rates, and practical reconfigurable phased arrays. While it is not 802.11ad-compliant, many of its features *resemble* those of 802.11ad.

Each X60 node consists of a mmWave transceiver system from NI [34] and a user-configurable phased antenna array from SiBeam. Transmissions take place over a 2 GHz wide channel, same as in 802.11ad. The PHY reference implementation supports 9 Single Carrier (SC) MCSs resulting in data rates from 300 Mbps to 4.75 Gbps, similar to those supported by the SC 802.11ad PHY layer. In contrast to COTS 802.11ad radios that use CSMA, X60 uses TDMA with 10 ms frames divided into 100 slots of 100 $\mu s$ each. A slot consists of 92 codewords, each of which has an attached CRC block. For our study, which only includes single-link experiments and hidden terminal scenarios, where CSMA is not useful, the use of TDMA instead of CSMA does not affect our results. Also, note that the structure of an X60 frame resembles an 802.11 aggregated frame (AMPDU), consisting of multiple packets, each with its own CRC.

The in-built phased array has 24 elements; 12 each for Tx and Rx. SiBeam's reference codebook defines 25 beam patterns that can be steered in real-time (electronic switching in $< 1\mu s$). The beams are spaced roughly 5° apart in their main lobe, thus spanning around 120° in the azimuth, from -60° to 60°. The 3 dB beamwidth ranges from 25° to 35°, hence, each beam's main lobe overlaps with several neighboring beams. The beam patterns feature large side lobes in addition to the central main lobe, similar to the beam patterns in COTS 60 GHz devices [54].

## 4.2 Environments and scenarios

We collected a *main* dataset by taking measurements in multiple environments with different characteristics in a campus building: an open lobby, a lab, a conference room, and three corridors of width 1.74 m, 3.2 m, and 6.2 m. Details about each environment can be found in Appendix A.2.1. We consider three typical scenarios capturing all the factors that can trigger BA or RA due to a drop in channel quality.

**Linear and/or angular displacement.** In each environment, we fixed the Tx position and orientation, we selected an initial Rx position, and then moved or rotated the Rx to cause different levels of signal attenuation due to increased distance, Tx/Rx misalignment, or both. In all the rotation experiments, we rotated the Rx from 0° to -90° and from 0° to 90° in steps of 15° (where 0° is the initial orientation at each position) facing the Tx. Details about each environment can be found in Appendix A.2.2.

**Blockage.** We hand picked a few representative positions in each environment and repeated the measurements by introducing human blockage on the LOS path between the Tx and the Rx, at 3 positions: 1) in the middle between the Tx and Rx, 2) near the Tx and 3) near the Rx.

**Interference.** We used a TP-Link Talon AD7200 router communicating with an Acer P446-M laptop as a hidden terminal. We placed it at different positions and tried different sectors to create 3 levels of interference: 1) High interference: the throughput of the X60 link drops by ~80%, 2) Low interference: throughput drops by ~20%, and 3) Medium interference: throughput drops by ~50%. We performed these measurements at the same locations as the blockage experiments.

**Table 1: Main/training dataset summary.**

| | Number of Cases | | | Number of Positions | | | | |
|---|---|---|---|---|---|---|---|---|
| | *Total* | *BA* | *RA* | *Total* | *Lobby* | *Lab* | *Conf.* | *Corridors* |
| *Displacement* | 479 | 380 | 99 | 94 | 22 | 13 | 10 | 49 |
| *Blockage* | 81 | 72 | 9 | 12 | 4 | 1 | 2 | 5 |
| *Interference* | 108 | 36 | 72 | 12 | 4 | 1 | 2 | 5 |
| *Overall* | 668 | 488 | 180 | 118 | 30 | 15 | 14 | 59 |

Table 1 provides a summary of the dataset, listing the number of measurement positions for each type of link impairment, the number of entries for each type of link impairment, and the number of cases where BA outperformed RA and vice versa (details in §5).

# 5 DATASET AND GROUND TRUTH

## 5.1 Collection methodology

We use the term *state* to describe every position, orientation, and the presence/absence of blockage or interference. We define the *initial state* as: the Rx position closest to Tx for each displacement scenario in the lobby, lab, and corridors; Rx position 0 for each displacement scenario in the conference room (Fig. 14c in Appendix A.2.1); the 0° Rx orientation for each rotation scenario; and the state before the introduction of blockage or interference for each blockage and interference scenario. All other states, at which the Rx position, Rx orientation, or the blockage or interference status is different from the initial state are called *new states*; these are the states where RA or BA (or both) are needed to repair the link.

At each state, we first performed a SLS to collect SNR measurements for all 625 ($25 \times 25$) beam pairs and selected the best beam pair based on SNR. *This process emulates BA using the naive $O(N^2)$ algorithm* described in §2. Then, for the best beam pair, we collected three 1 s PHY layer traces (SNR, Noise level, power delay profile (PDP), codeword delivery ratio (CDR)) and MAC throughput traces for each of the 9 supported MCSs. X60 logs all these metrics for every frame. We also measured offline the time-of-flight (ToF) for the chosen beam pairs at all positions. For all new states, we also collected PHY and throughput traces and ToF values for the beam pair that was the best at the corresponding initial state. *Searching over all the MCSs with the best current/initial beam pair and selecting the one with the highest throughput emulates RA after/before BA at the new state.* We confirmed experimentally that the average values of all the PHY and MAC layer metrics do not change drastically for durations of several seconds at a given position and for a given beam pair and MCS, since we keep the environment controlled.

Each dataset entry includes the change in the value of each PHY metric (details in §6.1) collected at that position before and after a link impairment, the initial best MCS, and a label specifying which of the two mechanisms (RA or BA) should be triggered, based on the ground truth, which we calculate using the measured throughputs after introducing the link impairment.

## 5.2 Ground truth

Selecting the best mechanism at a given scenario depends on the specific RA/BA algorithms used, the MAC/PHY protocol parameters that determine the overhead, as well as by the metric one wants to optimize. Once the RA/BA algorithms and the metric of interest are chosen, defining the ground truth for a given entry in the dataset boils down to "simulating" BA and RA using the logged SNR and throughput traces, and selecting as the ground truth winner for that entry the mechanism that optimizes the metric of interest. For example, if the metric of interest is throughput, then a first definition of the ground truth could be: *"perform RA when* $Th(RA) \geq Th(BA)$*, and BA otherwise"*, where $Th(RA)$ is the throughput achieved with RA, i.e., *the highest throughput among all the MCSs with the beam pair same as the highest-SNR beam pair in the initial state*; and $Th(BA)$ is the throughput achieved with BA, i.e., *the throughput with the highest-SNR beam pair, using the same MCS as the highest-throughput MCS in the initial state.*

**RA/BA subtleties.** In many entries of the dataset, the throughput after BA using the highest-throughput MCS in the initial state is zero, but lowering the MCS by 1 level results in non-zero throughput. It would be unrealistic to define $Th(BA) = 0$ in such cases, since any real MAC protocol would perform RA right after BA, if the first transmission with the new best beam pair failed. Following the guidelines in [14], we assume BA is *always* followed by RA. Hence, our goal essentially becomes to determine *whether RA alone is enough to optimize the metric of interest at a new state or BA should be triggered first, followed by RA.* We also assume that RA always starts at the best initial MCS and explores all the MCSs lower than that, sending one frame at each MCS. When RA is triggered first, this is expected, as the goal here is to repair a link, and hence a higher, less robust MCS is unlikely to help. When RA is triggered after BA, this assumption again makes sense, since BA typically discovers a longer path via reflection, which is unlikely to support an even higher MCS than before. Based on these assumptions, we redefine $Th(BA)$ *as the highest throughput among all the MCSs lower than or equal to the initial MCS using the highest-SNR beam pair.*

**Optimization metric.** Although throughput seems an obvious choice for the optimization metric, we note that BA uses *control frames* to test each beam pair. Hence, *throughput is always 0 during BA.* In contrast, RA *data frames* to test different MCSs. Hence, *throughput is suboptimal but not necessarily 0 during RA.* Consequently, another metric of interest is the *link recovery delay*, i.e., the delay from the moment a link breaks until we discover the first *working MCS.* Since 802.11 standards do not mandate a specific RA algorithm, different RA algorithms in the literature define a working MCS in different ways, e.g., as an MCS that yields a non-zero throughput or a throughput/loss rate above/below a threshold. Here, we define a working MCS as any MCS that satisfies two conditions: (1) $CDR > 10\%$ and (2) $Th > 150$ Mbps (50% of the PHY data rate of the lowest MCS).

In general, RA may be sufficient to quickly restore the link sub-optimally (and minimize the link recovery delay $D$), but BA may be required to discover the new optimal beam pair (and maximize throughput $Th$). Thus, we combine the two metrics in one utility metric $U$ as follows: $U = \alpha \cdot \frac{Th}{Th_{max}} + (1-\alpha) \cdot (1 - \frac{D}{D_{max}})$ (1), where $Th_{max}$ is the PHY data rate of the highest MCS. The worst-case delay $D_{max}$ is incurred when BA is the right choice and the optimal MCS is MCS 0 but RA is instead triggered first probing all the available MCSs (taking $N_{MCS} \cdot d_{fr}$ time, where $N_{MCS}$ is the number of available MCSs and $d_{fr}$ is the frame duration) without discovering a working MCS, then BA is performed (e.g., a SLS taking $d_{BA}$ time to complete), followed by another round of RA, which again probes all $N_{MCS}$ MCSs finally discovering MCS0, i.e., $D_{max} = N_{MCS} \cdot d_{fr} + d_{BA} + N_{MCS} \cdot d_{fr}$. The parameter $\alpha \in [0, 1]$ allows us to change the weight we assign to each metric.

Columns "BA" and "RA" in Table 1 show the number of cases where BA outperformed RA and vice versa, using the ground truth definition with $\alpha = 1$ for simplicity (i.e., maximizing throughput). As we can see, RA alone should be triggered in 27% of the cases (180/668) and BA should be triggered before RA in 73% of the cases. However, these numbers change if we consider each type of link impairment separately. Under displacement, RA outperforms BA in 21% of the cases, but under blockage only in 9/81 cases. On the other hand, under interference, RA is the preferred option in 67% of the cases. These differences motivate us to first study the problem separately under each link impairment type and then using the combined dataset.

# 6 LINK ADAPTATION USING PHY LAYER INFORMATION

In this section, we explore the use of PHY layer information to guide link adaptation.

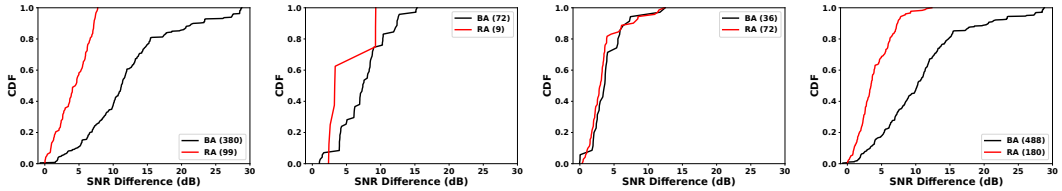
## 6.1 Metrics

We examine a number of PHY layer metrics and we explore whether each of them can predict the right adaptation mechanism in case of different link impairments. Some of these metrics are intuitively useful, while others have been used in previous link adaptation studies in legacy WiFi, e.g, [55]. For each metric, we plot in Figs. 4-9 the CDF of the metric values for all the cases where BA outperforms RA (denoted as "BA") and for all the cases where RA outperforms BA (denoted as "RA"), separately for each of the 3 datasets corresponding to the three link impairment scenarios in Table 1, and for the combined dataset. We again assume $\alpha = 1$ for simplicity. Our goal is to investigate whether we can identify clear thresholds for (some of) the metrics that allow us to separate the cases where each adaptation mechanism should be triggered.

*6.1.1 Displacement.* We define each metric and study its behavior using the "Displacement" dataset (the largest of the three datasets), where in the majority of the cases (79%) BA outperforms RA.

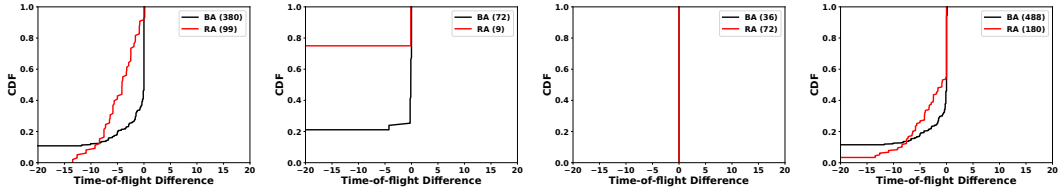
**SNR Difference.** We consider the difference between the SNR at the initial and the current state, each averaged over 1 s. As Fig. 4a shows, when the SNR drop is more than 7 dB, BA always outperforms RA. The cause of a high SNR drop is an angular displacement that results in beam misalignment; in such cases, lowering the MCS is not enough to repair the link. Using this threshold, we can classify 73% of the BA cases in this scenario. However, when the SNR drop is lower than 7 dB, the number of cases where each mechanism outperforms the other one is roughly the same.

**ToF Difference.** We consider the difference between the ToF at the initial and the current state. ToF increases with the distance and hence, one can expect a non-zero ToF difference in cases where



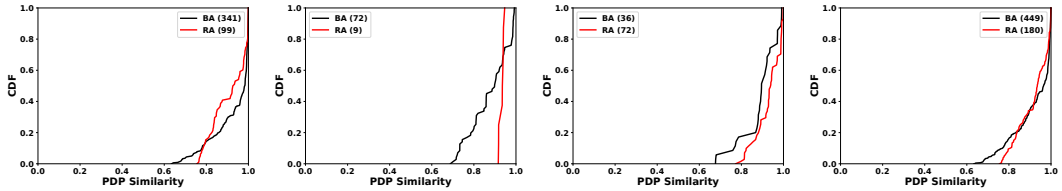

(a) Displacement. (b) Blockage. (c) Interference. (d) Overall.

**Figure 4: SNR Difference**



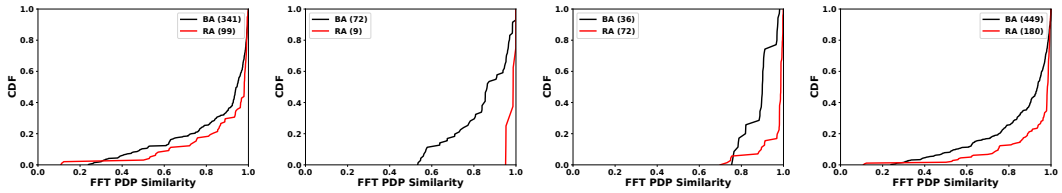

(a) Displacement. (b) Blockage. (c) Interference. (d) Overall.

**Figure 5: ToF Difference**




(a) Displacement. (b) Blockage. (c) Interference. (d) Overall.

**Figure 6: PDP Similarity**




(a) Displacement. (b) Blockage. (c) Interference. (d) Overall.

**Figure 7: CSI Similarity**

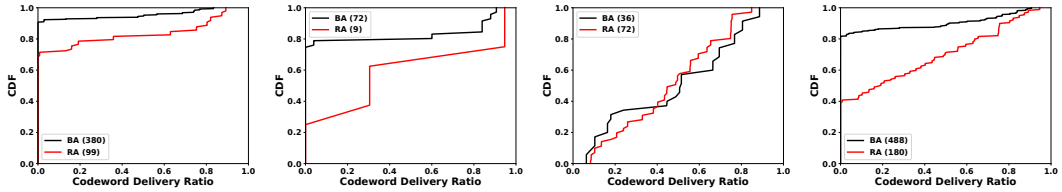
linear displacement is involved (backward, lateral, or diagonal motion) and zero difference in cases involving only rotation. We also note that X60 reports the ToF as infinity in cases of extremely weak signal (e.g., after a 90° rotation). Fig. 5a shows that in almost all cases where RA outperforms BA, the ToF difference is negative, corresponding to backward motion. In such cases, where the beams remain aligned, selecting a lower MCS is sufficient. On the other hand, when the ToF difference is 0 or infinity, BA is always needed to restore the link. However, for about 30% of the cases, BA outperforms RA even though the ToF difference is negative (and hence only backward motion is involved). This result agrees with our conclusion using COTS devices in §3, Fig. 3c; the LOS path is not always the best path.

**Noise Level Difference.** We define the Noise Level difference similar to SNR and ToF difference. We found that, when the noise level difference is higher than 7 dB, BA always outperforms RA (the figure is omitted due to space limit). However, this threshold allows us to only classify 65% of the RA cases.

**Multipath-related Metrics.** Researchers have proposed the use of Channel State Information (CSI) to guide RA in legacy OFDM-based WiFi systems and have shown that CSI-based algorithms [25, 55] outperform traditional algorithms. CSI captures fine-grained variations of the wireless channel due to multipath propagation, which cannot be captured by metrics such as SNR. *To our best knowledge, the use of CSI in RA for 60 GHz WLANs has not been explored before.*

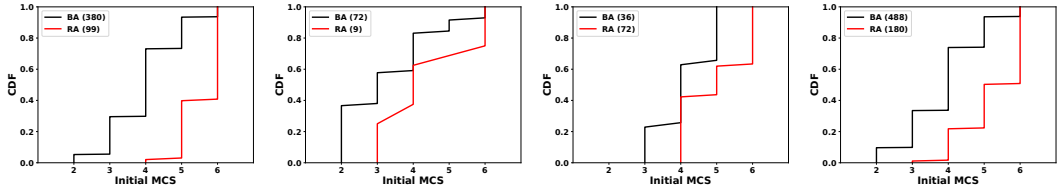
Since X60 only supports an SC PHY similar to COTS 802.11ad devices, we cannot measure directly CSI. Instead, we log PDP, which also captures the impact of multipath propagation, but in the time domain instead of the frequency domain. We also perform an FFT of the PDP to convert it from the time domain to the frequency domain and use it as an estimate of CSI. Following [55], for each of these metrics, we calculate the similarity between the two instances of the metric (at the initial and current state) in the form of the Pearson correlation coefficient.

Fig. 6a shows that PDP similarity cannot be used to predict the type of link adaptation. Interestingly, PDP exhibits very high similarity – at least 0.9 in 68% of the cases and always higher than 0.65, owing to the sparsity of 60 GHz channels, which results in limited multipath effects compared to sub-6 GHz frequencies [56, 57]. On the other hand, similarity values in the frequency domain exhibit much more diversity (Fig. 7a), ranging from 0.1 to 1. However, the two CDFs still exhibit a very large degree of overlap, which prevents us from defining a clear threshold to separate them.




(a) Displacement. (b) Blockage. (c) Interference. (d) Overall.

**Figure 8: Codeword Delivery Ratio (CDR)**




(a) Displacement. (b) Blockage. (c) Interference. (d) Overall.

**Figure 9: Initial MCS**

**Error/Delivery Rate.** State-of-the-art RA algorithms for legacy WiFi systems are often based on the subframe error rate (SFER), the fraction of successful MPDUs inside an AMPDU. Since the X60 PHY does not support frame aggregation, we use CDR, the fraction of successful codewords in a 10 ms X60 frame to approximate SFER in WiFi. Note that the length of an X60 frame is same as the maximum allowed AMPDU length in 802.11n/ac and the codeword size (180-1080 bytes for different MCSs) is similar to an MPDU size. Fig. 8a shows that CDR is 0 for about 90% of the BA cases and about 70% of the RA cases, and hence, it cannot be used alone to predict the correct adaptation mechanism.

**Initial MCS.** We also explore the use of the best initial MCS as a potential metric. A high initial MCS suggests a very strong link. In such cases, a small linear or angular displacement might not cause a large drop in channel quality and dropping the MCS by 1-2 levels might be more effective than using a new beam over a NLOS path. In contrast, if the initial MCS is already low, there is not enough margin for RA to repair the link and switching to a different path via BA might be a better solution. Fig. 9a confirms our intuition. When the initial MCS is 2, 3, or 4, BA almost always outperforms RA. On the other hand, the initial MCS is 5 or 6 in 98% of the cases where RA outperforms BA but also in 23% of the cases where BA outperforms RA. Hence, initial MCS alone cannot always predict the correct adaptation mechanism.

*6.1.2 Blockage.* Interestingly, Table 1 shows that, regardless of the environment or blockage position, BA outperforms RA in all but 9 cases. This does not necessarily mean that we fully blocked the LOS in each case. For example, Fig. 4b shows that the SNR drop spans a range of 1-15 dB and Fig. 5b shows that we were able to measure ToF in 73% of the cases. Nonetheless, even in partial blockage, our results show that BA is almost always the best option. Figs. 4b-9b show that, although the behavior of most metrics under blockage is different from the behavior under displacement, there is still a large overlap between the values of each metric observed when RA outperforms BA and those observed when BA outperforms RA, making it hard to identify appropriate thresholds.

*6.1.3 Interference.* Compared to displacement and blockage, interference in 60 GHz has received much less attention; there is only a small number of experimental studies [16, 46, 56, 66] and only [66] has briefly investigated the interplay between interference and link adaptation. The first observation from the interference dataset is that RA is most often the preferred mechanism, outperforming BA in 72/108 cases. More importantly, Figs. 4c-9c show a very different picture for most metrics under interference compared to displacement or blockage scenarios. However, the overall conclusion remains the same; it is not possible to identify appropriate thresholds for any metric in interference scenarios.

*6.1.4 Combined Dataset.* The different behavior of the metrics under each type of link impairment makes it even harder to define generic, scenario-independent thresholds to easily separate the cases where each of the two adaptation mechanisms outperforms the other, as is evident from Figs. 4d-9d. While for certain metrics one can still identify thresholds, similar to the displacement scenarios, these thresholds have been shifted compared to Figs. 4a-9a, and they separate a smaller fraction of cases. For example, the SNR difference threshold increases now to 12 dB from 7 dB in Fig. 4a and it can only identify 30% of the cases where BA outperforms RA (compared to 73% in Fig. 4a).

Overall, Figs. 4-8 show the fundamental reason why selecting the right link adaptation mechanism is hard in practice. Adaptation is triggered to repair an impaired link. All types of link impairment (displacement, blockage, interference) manifest as a drop in the link quality, which is reflected via a change to one or more PHY layer metrics used to measure link quality (e.g., a drop in the SNR or the CDR, an increase in the noise level, a drastic change to the CSI, etc.). Our study showed that all these metrics are affected in a similar way when either RA or BA is the preferred adaptation mechanism. The use of phased arrays with imperfect beam patterns by today's COTS devices and the complexity of typical indoor environments further exacerbate the problem, as we saw in §3. As a result, it becomes hard to design simple heuristics relying on any of these metrics to select the right action, even in seemingly simple scenarios.

## 6.2 Learning-based Adaptation

While no single metric can be used alone to guide link adaptation, different metrics can be used to separate different subsets of the two classes based on thresholds. Since such thresholds and the order in which they should be applied are hard to identify by simple

**Table 2: Testing Dataset Summary.**

| | Number of Cases | | | Number of Positions | | |
|---|---|---|---|---|---|---|
| | *Total* | *BA* | *RA* | *Total* | *Building 1* | *Building 2* |
| *Displacement* | 165 | 129 | 36 | 34 | 23 | 11 |
| *Blockage* | 27 | 24 | 3 | 4 | 2 | 2 |
| *Interference* | 36 | 12 | 24 | 4 | 2 | 2 |
| *Overall* | 228 | 165 | 63 | 42 | 27 | 15 |

inspection of the dataset and the CDFs in Figs. 4d-9d, we turned to ML-based approaches. We tried 3 popular classical ML models, which are most suited for our 2-class classification problem – decision trees (DT), random forests (RF), and support vector machines (SVM). In the case of DT and RF, we tried two impurity measures: Gini index and entropy. We also limited the maximum depth of the trees generated by both algorithms to reduce overfitting. For SVM, we tried both linear and non-linear classification metrics and different regularization parameters. We also tried a deep neural network (DNN) model – a fully connected dense network with 4 dense layers. Rectified linear (relu) activation was used in the first 3 layers and sigmoid activation was used in the last layer. In order to reduce overfitting to the data, we explored different techniques present in the literature. Among these techniques, inclusion of Dropout [52] after each layer gave the best results.

**5-fold cross validation.** We used all the metrics discussed in §6.1 as input to the ML models. We then ran a stratified 5-fold cross validation on the entire dataset, and calculated the accuracy and the weighted F1 score. We repeated this process 500 times with random splits of the dataset. *All four models achieved very high average accuracy and F1 scores.* The accuracy/F1 score for the best combination of parameters with each model was 95%/95% with DT, 98%/98% with RF, 91%/91% with SVM, and 95%/90% with DNN.

**Accuracy with a different dataset.** Several of the metrics capture properties of the multipath channel structure, which is heavily affected by the environment. To evaluate whether the ML models, which are trained on a dataset collected in one building, can accurately predict the correct adaptation mechanism in different buildings, we collected a new dataset (*testing* dataset in Table 2) in two different buildings. In Building 1, we conducted measurements in a long 2.5 m wide corridor with the Rx at several distances away from the Tx. This building is much older than the building where we collected our main dataset, with walls of different material, and fewer reflective surfaces. In Building 2, we conducted measurements in a wide open area, much larger than the lobby in Fig. 14a. We trained the ML models on the main dataset and tested them on the testing dataset. Both the accuracy and F1 score dropped but still remained at satisfactory levels (85%/85% with DT, 88%/88% with RF and SVM, 83%/76% with DNN).

**Metric importance.** We calculated the Gini importance of each metric using the testing dataset. The results are shown in Table 3. We observe that the initial MCS and SNR have the highest importance (above 0.2), followed by the Noise Level. On the other hand, as expected from Fig. 6, PDP has the lowest importance. Nonetheless, we observe that no metric has a very high value, suggesting that all metrics are useful in our classification problem. We also want to point out that the metric selection depends on the used hardware, as the values of some of the metrics are hardware dependent. For example, we found out that the noise level values span a large range with X60 even in the absence of interference. In practice, chipset vendors might exclude some of the metrics for different types of hardware, based on the importance of each of them.

**Table 3: Gini importance.**

| SNR | ToF | Noise Level | PDP | CSI | CDR | Initial MCS |
|---|---|---|---|---|---|---|
| 0.215 | 0.08 | 0.16 | 0.06 | 0.12 | 0.125 | 0.26 |

## 7 LiBRA DESIGN

While the accuracy drops when the models are trained and tested in different buildings, triggering the wrong adaptation mechanism does not always have the same performance impact in practice. The cost depends on two factors: the overhead of triggering the wrongly predicted mechanism, which can delay restoring the link (e.g., the overhead of a full SLS in the case of wrongly triggering BA or the overhead of trying various MCSs in the case of wrongly triggering RA) and the overhead of sending data at a suboptimal PHY data rate after restoring the link. The impact of each of these factors depends on the specific RA and BA algorithms used, the MAC and PHY layer parameters (e.g., number of beam pairs, number of supported MCSs and their data rates, frame duration, etc.), the data flow duration, and the optimization metric.

To study the impact of triggering the wrong adaptation mechanism on the overall performance under practical settings, in this section, we design ***LiBRA*** (described in Algorithm 1 in §A.1), a *practical, standard-compliant* **L**earn**i**ng-based **B**eam and **R**ate **A**daptation framework. *LiBRA* leverages PHY layer information to determine (i) when to trigger link adaptation and (ii) which of the two adaptation mechanisms should be triggered. All the PHY layer metrics used by *LiBRA* are available in the firmware of both legacy and 60 GHz COTS devices; hence, 60 GHz chipset vendors can easily implement *LiBRA* at the firmware and/or the driver level. *LiBRA* optimizes the utility metric defined in Eqn. (1) in §5.2; the network operator can select the value of $\alpha$ depending on the metric (throughput vs. delay) they prefer to optimize. The design of a practical link adaptation approach has to address the following three issues:

**1) When to trigger link adaptation?** While the ML models in §6.2 determine which of the two mechanisms should be triggered after a link impairment, in practice we also need a method to determine whether adaptation is needed to deal with a link impairment. We tried again a learning based approach. We augmented our training and testing datasets in Tables 1 and 2, respectively, with new entries, one for each new state, where we considered the first 1 s of PHY layer and throughput traces with the best beam pair and MCS at that state as an initial state and the second 1 s trace as the new state. We then trained the RF ML model from §6.2 using three classes – BA, RA, and NA (No Adaptation). Our NA entries include static LOS, blockage, and interference scenarios. The accuracy of the RF model using 5-fold cross validation on the training dataset was 98% and using the testing dataset was 94%. *We thus use this 3-class model in the design of* LiBRA.

**2) Length of observation window.** In training and testing the ML models until now, we used traces of length equal to 2 s (1 s before and 1 s after a link impairment). In reality, such a long observation window before triggering adaptation is not practical. Hence, we retrained and tested our 3-class RF model using traces of total duration equal to 40 ms. The accuracy dropped only by 3 percentage points, using the test dataset. Hence, LiBRA *makes decisions every*

*20 ms (or 2 frames in X60)*, using the values of the metrics averaged over two 20 ms observation windows. In practice, the observation window can get shorter depending on the protocol frame duration; e.g., in 802.11ad, where the max frame duration is 2 ms, *LiBRA* would make decisions every 4 ms. The overhead of running the ML model is negligible in modern smartphones equipped with GPUs (0.5 ms in the ROG phone [6], the first 802.11ad-enabled smartphone).

Note that our approach of ignoring long-term history in making adaptation decisions is consistent with state-of-the-art RA designs for legacy WiFi [47, 63], which also use observations windows ranging from a few ms up to a few tens of ms. However, in the case of 60 GHz, longer observation windows may have some benefits, e.g., they may allow the transmitter to learn blockage patterns and make better decisions in the future. We believe that learning link status patterns over longer periods of time is an interesting avenue for future investigation, but is out of scope of this work.

**3) Tx- or Rx-initiated adaptation?** While a Rx-initiated approach appears to be the natural choice, since all the PHY layer metrics are collected on the Rx side, we note that all legacy RA and BA algorithms are Tx-initiated. Hence, an Rx-initiated approach would require new control frames for the Rx to notify the Tx to trigger a rate search or a SLS. Since we target a standard-compliant design, we selected a Tx-initiated approach. A challenge with this approach is to make the PHY layer metrics available on the Tx side in a standard-compliant manner that prevents the use of new control frames. We address this challenge by *leveraging the 802.11 ACKs and channel reciprocity*, similar to [23, 55]. However, a second challenge appears in the case of a lost frame; the Rx will not send back an ACK and hence, the Tx will never update its PHY layer metrics. We note that a missing ACK is a clear indication that the channel has worsened and link adaptation is required; the challenge is in selecting RA or BA. Using our training dataset, we observed that when the current MCS is lower than 6, BA is the right mechanism 92% of the time. Hence, *LiBRA* always triggers BA in these cases. On the other hand, with MCS 6 or higher, BA is the right choice 48% of the time and RA 52% of the time. Here, our choice depends on the BA overhead. We trigger BA first when the BA overhead is low (up to a few ms) and RA first otherwise.

**Adaptation algorithms.** *LiBRA* is generic and works with a variety of RA and BA algorithms from the literature. Our goal is not to design new optimal RA/BA algorithms but to demonstrate the benefits of triggering the right mechanism at each scenario and the efficiency of leveraging learning-based models and PHY layer information. In §8, we evaluate *LiBRA* with different standard-compliant BA algorithms that incur different amount of overhead.

In contrast to BA, which is only triggered to repair a link, a practical RA algorithm also has to occasionally explore higher working MCSs than the one in use. Here, we design a simple frame-based RA algorithm inspired by legacy algorithms [14, 47, 62, 63]. However, we note that other algorithms could be used instead. When RA is triggered to deal with a link impairment, it starts at the current MCS and probes all the lower MCSs (by sending one AMPDU at each of them) until it finds the highest-throughput working MCS. If no working MCS is found, BA is triggered, followed by another round of RA at the new best beam, starting at the same MCS that was in use before adaptation was triggered. To explore higher working MCSs, *LiBRA* follows an approach similar to [63]. It estimates the average CDR over an interval $T$ and probes the immediately higher MCS if $CDR > CDR_{ORI}$, where $CDR_{ORI}$ is a threshold calculated as described in [63]. To reduce the probing frequency to MCSs that consistently offer lower throughput, we use an adaptive probing interval $T$ inspired by [47]: $T = T_0 \cdot min(2^k, 2^5)$, where $k$ denotes the number of failed probes at the higher MCS (yielding throughput lower than the current MCS) and $T_0$ is the minimum probing interval, equal to 5 frames in our implementation (50 ms in X60 or 10 ms in 802.11ad).

**ML model training.** In §6.2, we showed that the RF model retains satisfactory accuracy, when tested in environments with very different characteristics compared to the environment used for training. Hence, we expect that *offline* training will be sufficient for *LiBRA*, as long as a comprehensive dataset, including all three types of link impairment – linear and angular displacement, blockage, interference – is used for training. On the other hand, as we mentioned in §6.2, vendors might have to perform separate training for different types of hardware (e.g., different AP models).

## 8 LiBRA EVALUATION

In this section, we evaluate *LiBRA* in diverse indoor scenarios. Although the PHY layer information used by *LiBRA* is available at the firmware level of COTS 60 GHz devices (§7), 60 GHz drivers currently do not expose this information to the user. This is a common practice even with legacy WiFi chipsets, with two notable exceptions [26, 64]. Hence, we are not able to implement *LiBRA* in COTS 802.11ad hardware. On the other hand, the X60 nodes do not support real-time bidirectional communication. Since *LiBRA* relies on ACKs to extract PHY layer information, implementing *LiBRA* on X60 is not feasible either. Finally, an implementation on a different SDR-based platform, e.g., OpenMili [68] or a narrowband platform [24, 56], would compromise realism, as no platform other than X60 provides performance and MAC/PHY features commensurate to those of 802.11ad. Since a realistic implementation is not feasible on any available hardware platform, we resort to trace-based simulation with realistic PHY/MAC parameters for the evaluation of *LiBRA*, which is a standard practice in the design of mmWave systems [23, 24, 56, 57, 61, 69, 70].

We are not able to directly compare *LiBRA* against MOCA, the non-standard-compliant approach from [24], for two reasons: (i) Beam Sounding frames in MOCA are sent at MCS 0, (27.5 Mbps) – the most robust MCS, used only for control frames in 802.11ad. In X60, the lowest supported MCS yields a data rate of 300 Mbps, very close to MCS 1 (385 Mbps) in 802.11ad. Sending the sounding frames at that MCS would result in a significantly lower delivery probability; (ii) X60 does not offer a multi level codebook with different beamwidths, which is a key component of MOCA's beamwidth adaptation algorithm. However, in our recent work [9], we showed that maintaining a failover sector, as proposed in [24], does not work in scenarios involving angular displacement between the Tx and Rx, which are very common in practice, and SNR-based RA performs poorly in real-world indoor settings.

### 8.1 Evaluation methodology

**Protocol parameters.** The time to perform BA depends on the beamwidth (which determines the number of beams to test) and the BA algorithm. We consider four realistic values in our evaluation:

0.5 ms, 5 ms, 150 ms, and 250 ms. For the first two values, we used Eqn. (2) from [24], which calculates the BA overhead in case of 802.11ad COTS devices ($O(N)$ search algorithm with quasi-omni reception), with a 30° beamwidth – used in X60 and most commercial devices today – and a 3° beamwidth – the minimum allowed by 802.11ad. For the last two values, we assumed directional reception and the $O(N^2)$ search algorithm that trains both Tx and Rx beams and used Fig. 11 in [56] with a 9°/7° beamwidth. We used $\alpha = 0.7$ with the first two BA overhead values; since the link recovery delay in these cases is expected to be low, we give more weight to throughput. In contrast, when the BA overhead is high (150 or 250 ms), we use $\alpha = 0.5$, increasing the delay weight.

The time to perform RA to restore a link is given by the product of the number of MCSs traversed while searching for the highest-throughput working MCS and the maximum frame aggregation time (FAT) – recall that RA sends one aggregated frame at each tested MCS. We consider 2 different FAT values: 2 ms – the maximum value allowed in 802.11ad – and 10 ms – the maximum value allowed in 802.11ac, also used in X60.

**Metrics.** We compare the amount of data delivered with *LiBRA* against the amount of data delivered by an oracle solution Oracle-Data, which always triggers the right adaptation mechanism that maximizes the total amount of bytes delivered. This metric depends on both the throughput and the link recovery delay, as well as on the flow duration. For a short flow duration, it might be preferable to start sending data quickly at a suboptimal working MCS instead of looking for the optimal beam pair/MCS configuration. We also compare the link recovery delay alone against another oracle solution Oracle-Delay, which always triggers the right adaptation mechanism that minimizes the link recovery delay. We include two more heuristics in our study: always performing RA first, which is what all COTS devices do today; and always performing BA first which is suggested in [14]. Both heuristics trigger adaptation when the current MCS becomes non-working (§5.2). Note that for all algorithms, we take into account the BA or RA overhead in our calculations. The oracles always incur only one of the two overheads in each case, while the other three algorithms may incur both overheads in case of misprediction. Also, note that all algorithms (including the oracle solutions) use the same mechanism as *LiBRA* to probe higher rates periodically. The oracles make optimal decisions only with respect to restoring a link.

## 8.2 Single link impairment scenarios

We first consider simple scenarios, where there is only one link impairment, using our combined dataset from Buildings 1 and 2. We consider two data flow lengths: 1 s and 0.4 s.

Figs. 10a-10g plot the CDFs of the difference between the bytes delivered by Oracle-Data and each other algorithm for all combinations of the different RA/BA overheads and for the 2 flow durations. For the 1 s flow duration, we make the following observations: (1) *LiBRA* performs very close to the oracle in all cases and much better than the two heuristics in most cases. With a FAT of 2 ms, it delivers the same number of bytes as the oracle in about 85% of the cases. In contrast, "BA First" delivers the same number of bytes as the oracle in only 70-81% of the cases, and its efficiency worsens as the BA duration increases. "RA First" performs even worse, delivering the same number of bytes as the oracle in only 50-58% of the cases. With a FAT of 10 ms, the amount of bytes delivered by *LiBRA* is within 10 MB from the oracle in 80-84% of the cases. The same number is only 71-81% for "BA First" and 26-35% for "RA First". (2) The maximum loss for *LiBRA* is always much lower compared to both "RA First" and "BA First" for low BA overhead (0.5 and 5 ms) and within 100 MB from "BA First" in the case of very long BA duration (150 and 250 ms). (3) The FAT has a smaller impact on the performance compared to the BA duration.

With a short flow of 0.4 s, we make similar observations with the exception of the worst case performance with *LiBRA*, which is never lower than with "BA First" or "RA First". Overall, the flow duration has the lowest impact on "BA First", which triggers a constant overhead first (the overhead of a full SLS) but always discovers the optimal configuration. On the other hand, the impact of flow duration is highest on "RA First", which often settles down to a working but suboptimal MCS. The impact is particularly pronounced in the case of long flows, where the data loss due to the use of a suboptimal MCS is larger.

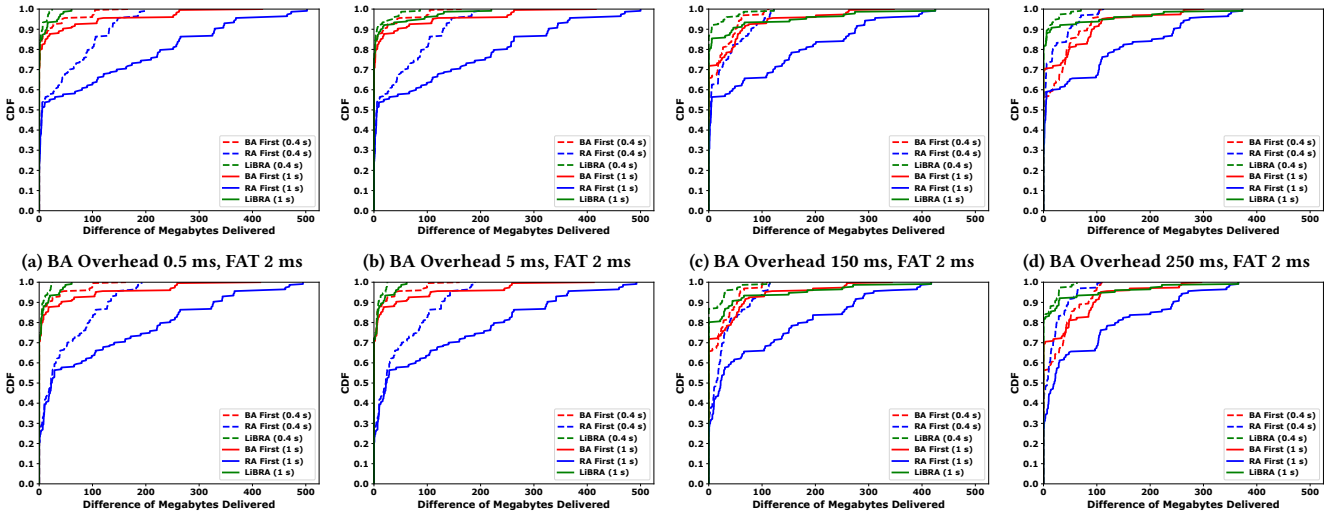
Figs. 11a-11g plot the CDFs of the difference between the recovery delay with the Oracle-Delay and each other algorithm. We observe that the recovery delay is the longest with "RA First" for low BA duration (0.5 ms, 5 ms) and with "BA First" for high BA duration (150 ms, 250 ms). In contrast, *LiBRA* strikes a balance, achieving within 5 ms from the optimal delay in 57-98% of the cases with all parameter combinations and long delays in a much smaller fraction of cases compared to "BA First". For example, with a 250 ms BA overhead, the median delay difference is 0 ms with *LiBRA* but more than 200 ms for "BA First".

Overall, these figures show that for today's 802.11ad devices that have only a few sectors and employ quasi-omni reception, limiting the BA overhead to a few ms [49, 60], "BA First" is a good choice, albeit its worst case performance is lower than *LiBRA*'s. The real benefit of *LiBRA* will become more prominent in the future, as the number of available sectors increases and devices start employing directional reception to improve the range or to limit interference in dense deployments [30, 66] or in the case of MIMO [23].

## 8.3 Multiple link impairment scenarios

We now evaluate *LiBRA* in scenarios involving multiple link impairments. Each scenario consists of 10 time segments, each of varying duration from 300 ms to 3 s. We consider 4 types of scenarios: (i) Mobility: We emulate mobility by moving the Rx at the beginning of each segment introducing differing degrees of linear and/or angular displacement, (ii) Blockage: we alternate segments of human blockage of random duration at random positions between the Tx and Rx and segments of clear LOS, (iii) Interference: we alternate segments of varying levels of interference and segments of clear channel, and (iv) Mixed: a combination of the above three. We collected 10 300-s PHY layer and throughput traces for each segment involved in each scenario. We then generated 50 random timelines of 3-30 s duration for each type of scenario, by choosing uniformly randomly a time between 300 ms and 3 s as the time to spend at each of the 10 segments. We only show the results for two BA overhead values (0.5 ms and 250 ms) due to space limits.

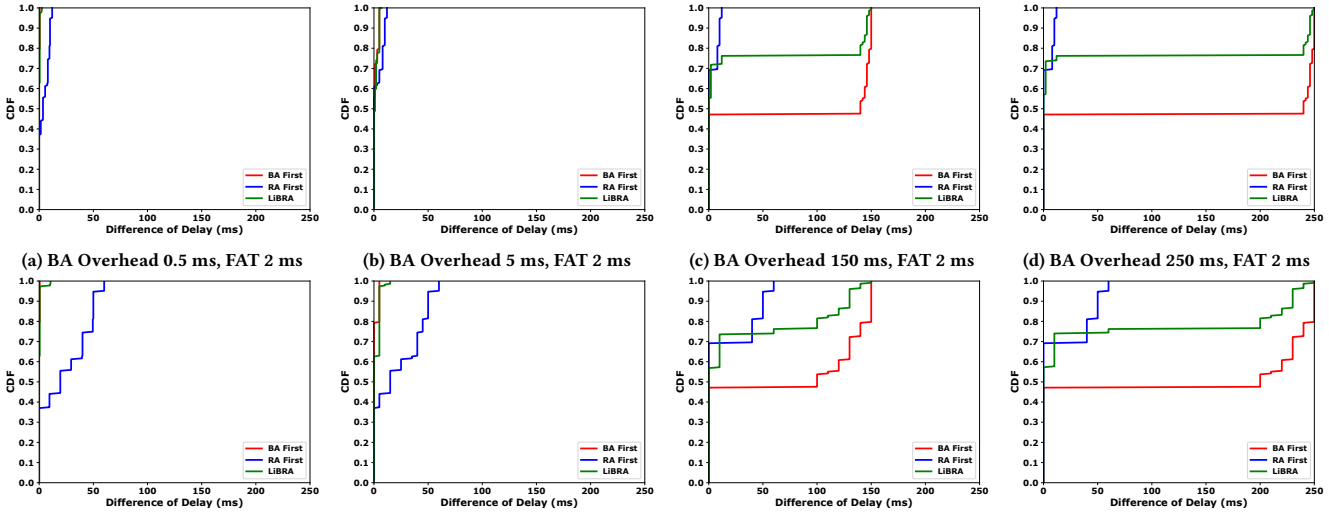
Fig. 12 plots the percentage of bytes delivered by each algorithm compared to the bytes delivered by Oracle-Data and Fig. 13 plots the difference between the average link recovery delay (the sum of




(a) BA Overhead 0.5 ms, FAT 2 ms (b) BA Overhead 5 ms, FAT 2 ms (c) BA Overhead 150 ms, FAT 2 ms (d) BA Overhead 250 ms, FAT 2 ms

(e) BA Overhead 0.5 ms, FAT 10 ms (f) BA Overhead 5 ms, FAT 10 ms (g) BA Overhead 150 ms, FAT 10 ms (h) BA Overhead 250 ms, FAT 10 ms

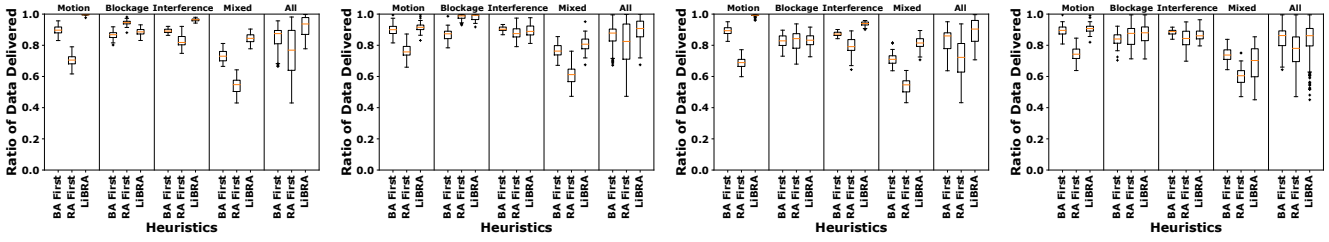
**Figure 10: Single link impairment: CDFs of the difference between the bytes delivered with Oracle-Data and each of the other three algorithms.**




(a) BA Overhead 0.5 ms, FAT 2 ms (b) BA Overhead 5 ms, FAT 2 ms (c) BA Overhead 150 ms, FAT 2 ms (d) BA Overhead 250 ms, FAT 2 ms

(e) BA Overhead 0.5 ms, FAT 10 ms (f) BA Overhead 5 ms, FAT 10 ms (g) BA Overhead 150 ms, FAT 10 ms (h) BA Overhead 250 ms, FAT 10 ms

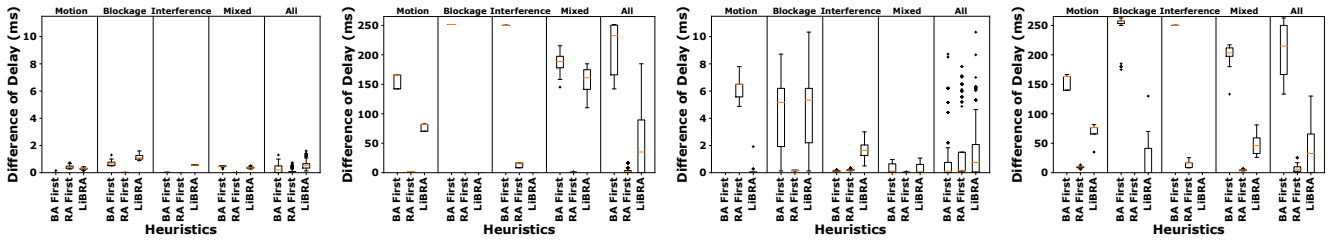
**Figure 11: Single link impairment: CDFs of the difference between the recovery delay with the Oracle-Delay and each of the other three algorithms.**




(a) BA Overhead 0.5 ms, FAT 2 ms (b) BA Overhead 250 ms, FAT 2 ms (c) BA Overhead 0.5 ms, FAT 10 ms (d) BA Overhead 250 ms, FAT 10 ms

**Figure 12: Multiple link impairments: percentage of bytes delivered by each algorithm compared to Oracle-Data.**

link recovery delays for each timeline divided by the number of link breaks) with Oracle-Delay and each of the other three algorithms. The results are plotted in the form of boxplots for all 50 timelines for each type of scenario separately and for all scenarios.




(a) BA Overhead 0.5 ms, FAT 2 ms (b) BA Overhead 250 ms, FAT 2 ms (c) BA Overhead 0.5 ms, FAT 10 ms (d) BA Overhead 250 ms, FAT 10 ms

**Figure 13: Multiple link impairments: difference between the average link recovery delay with the Oracle-Delay and each of the other three algorithms.**

**Table 4: VR stall duration (ms)/number of stalls.**

| BA Overhead, FAT | BA First | RA First | *LiBRA* | Oracle-Data | Oracle-Delay |
|---|---|---|---|---|---|
| **0.5 ms, 2 ms** | 16/46.4 | 16/97.5 | 16/0.1 | 0/0 | 16/46.5 |
| **0.5 ms, 10 ms** | 16.3/54.1 | 17.3/105.6 | 16.1/10.5 | 16/5.7 | 16.7/36.6 |
| **250 ms, 2 ms** | 49/51.4 | 21.7/97.3 | 240/6.1 | 236.7/6.1 | 21.4/97.3 |
| **250 ms, 10 ms** | 49.7/53.3 | 22.5/105.2 | 94.9/22.5 | 220.9/6.9 | 25/68.3 |

We observe that *LiBRA* delivers more bytes than "BA First" and "RA First", when we consider all four types of scenarios ("All" in Figs. 12a-12d) for any combination of BA overhead and FAT. In the median case, *LiBRA* delivers 90-95% of the bytes delivered with Oracle-Data; the same numbers are 90-92% for "BA First" and 71-82% for "RA First". When we look at different types of scenarios, we observe that, the Mixed type is, as expected, the most challenging type for all three schemes. Overall, *LiBRA* outperforms the two heuristics for the majority of scenarios and BA overhead/FAT combinations, delivering in the median case as high as 100% of the bytes (Motion, Figs. 12a, 12c) and never below 70% (Mixed, Fig. 12d). The numbers are lower for "BA First" and much lower for "RA First" (only 55% for Mixed in Figs. 12a, 12c).

Figs. 13a-13d show that *LiBRA* also keeps the average link recovery delay low, which is important for interactive applications that require both Gbps data rates and low delay (e.g., VR, UHD streaming, Miracast). The median recovery delay with *LiBRA* is at most 35 ms when considering all scenarios and it only takes excessive values (150 ms) in the Mixed scenarios in Fig. 13b. "BA First" has a very low link recovery delay (below 1 ms) when the BA overhead is low, but unacceptably high when the BA overhead grows (more than 170-250 ms in the median case), confirming again that it is a good short-term solution for today's 802.11ad devices but not for future ones. On the other hand, "RA First" always recovers the link fast, although at the cost of suboptimal performance in terms of bytes delivered, as we saw in Fig. 12.

## 8.4 *LiBRA* with a real application

We finally evaluate *LiBRA* using a real high-bandwidth, latency-sensitive application: VR at 60 FPS and 8K resolution.[2] We used a 30 s Viking Village [3] scene based on a specific trajectory. Since the application's bandwidth demand is no more than 1.2 Gbps, it did not make sense to perform our evaluation using the actual throughputs provided by X60, which achieves 4.75 Gbps at its highest supported MCS; all three heuristics were able to easily support this demand. For a meaningful evaluation, we scaled down the throughputs logged with each MCS in our traces to what one would expect from 802.11ad COTS devices (which achieve up to 2.4 Gbps when they are placed right in front of the AP [17, 48, 49]) using the same modulation and coding rate. Also, we used only the mobility traces from §8.3 in our evaluation, since one would not expect any external blockage or interference while playing a VR game.

Table 4 shows the average stall duration and average number of stalls for the 50 timelines involving mobility. We again show results for only 2 BA overhead values (0.5 ms, 250 ms) due to space limits. We observe that when BA overhead is low (0.5 ms), *LiBRA* has similar stall duration to "BA First" and "RA First" but a much smaller number of stalls. When the BA overhead is high (250 ms), *LiBRA*'s stall durations are longer but again the number of stalls is much smaller, resulting overall in better user QoE.

Interestingly, none of the two oracles can achieve optimal performance. Oracle-Data incurs the smallest number of stalls for all four combinations of BA overhead and FAT, but the average stall duration with high BA overhead is similar to or much longer than *LiBRA*'s. On the other hand, Oracle-Delay keeps the average stall duration low, but incurs a larger number of stalls. This result suggests that real applications often have conflicting requirements in terms of throughput and delay, and simultaneously optimizing those metrics can be hard in practice. *LiBRA*'s design strikes a good balance between the two metrics resulting in good user QoE.

## 9 CONCLUSION

We conducted one of the first extensive experimental studies of RA and BA in 60 GHz WLANs. Using experiments with COTS devices, we showed that selecting the right mechanism under a given scenario is far from trivial and simple heuristics can lead to wrong decisions even in seemingly simple scenarios. We then explored for first time the feasibility of leveraging PHY layer information and ML to guide link adaptation. Finally, we designed *LiBRA*, a practical, standard-compliant link adaptation framework that leverages ML and PHY layer information to determine when to trigger link adaptation and which adaptation mechanism to use. Via extensive trace-based simulations, we showed that *LiBRA* strikes a balance between throughput and link recovery delay, performing close to an oracle solution, and outperforming significantly two simple heuristics used by COTS devices.

## ACKNOWLEDGMENTS

We thank our shepherd and the anonymous reviewers for their valuable comments. This work was supported in part by the National Science Foundation grants CNS-1553447 and CNS-1801903.

[2]We considered 8K VR, because 4K VR does not demand throughput more than 300 Mbps, which can be supported even by legacy WiFi [41, 44].

# REFERENCES


[1] [n.d.]. Acer TravelMate P446-M. https://www.acer.com/ac/en/US/content/professional-series/travelmatep4

[2] [n.d.]. Minstrel Rate Adaptation Algorithm Documentation. http://madwifiproject.org/browser/madwifi/trunk/ath_rate/minstrel/minstrel.txt

[3] [n.d.]. *Viking village*. https://www.assetstore.unity3d.com/en/#!/content/29140

[4] 2016. TP-Link Talon AD7200 Multi-Band Wi-Fi Router. http://www.tp-link.com/us/products/details/cat-5506_AD7200.html

[5] 2017. Netgear Nighthawk X10 Smart WiFi Router. https://www.netgear.com/landings/ad7200/

[6] [Online]. *ASUS Republic of Gamers (ROG) Phone*. https://www.asus.com/us/Phone/ROG-Phone/

[7] online. Intel Tri-Band Wireless-AC 18265. https://www.intel.com/content/www/us/en/wireless-products/tri-band-wireless-ac-18265.html.

[8] Saeed Abdallah and Steven D. Blostein. 2016. Joint rate adaptation, frame aggregation and MIMO mode selection for IEEE 802.11ac. In *Proc. of IEEE WCNC*.

[9] Shivang Aggarwal, Urjit Satish Sardesai, Viral Sinha, and Dimitrios Koutsonikolas. 2020. An Experimental Study of Rate and Beam Adaptation in 60 GHz WLANs. In *Proc. of ACM MSWiM*.

[10] Ahmed Alkhateeb, Sam ALex, Paul Varkey, Ying Li, Qi Qu, and Djordje Tujkovic. 2019. Deep Learning Coordinated Beamforming for Highly-Mobile Millimeter Wave Systems. *ArXiv e-prints* (2019). arXiv:1804.10334v3

[11] Ahmed Alkhateeb, Omar El Ayach, Geert Leus, and Robert W Heath. 2014. Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems. *IEEE Journal of Selected Topics in Signal Processing* 8, 5 (2014), 831âĂŞ846.

[12] Ahmed Alkhateeb, Iz Beltagy, and Sam Alex. 2018. Machine Learning for Reliable mmWave Systems: Blockage Prediction and Proactive Handoff. In *Proc. of IEEE GlobalSIP*.

[13] Muhammad Alrabeiah and Ahmed Alkhateeb. 2019. Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6GHz Channels. *ArXiv e-prints* (2019). arXiv:1910.02900v3

[14] Qualcomm Inc. Amichai Sanderovich. 2016. Techniques For Rate Selection In Millimeter-Wave Communication Systems. US Patent 9,312,985.

[15] Arash Asadi, Sabrina MÃijller, Gek Hong (Allyson) Sim, Anja Klein, and Matthias Hollick. 2018. FML: Fast Machine Learning for 5G mmWave Vehicular Communications. In *Proc. of IEEE INFOCOM*.

[16] Hany Assasa, Swetank Kumar Saha, Adrian Loch, Dimitrios Koutsonikolas, and Joerg Widmer. 2018. Medium Access and Transport Protocol Aspects in Practical 802.11ad Networks. In *Proc. of IEEE WoWMoM*.

[17] Ghufran Baig, Jian He, Mubashir Adnan Qureshi, Lili Qiu, Guohai Chen, Peng Chen, and Yinliang Hu. 2019. Jigsaw: Robust Live 4K Video Streaming. In *Proc. of ACM MobiCom*.

[18] John C. Bicket. 2005. *Bit-rate Selection in Wireless Networks*. Master's thesis. MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

[19] Joseph Camp and Edward Knightly. 2010. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-Layer Implementation and Experimental Evaluation. *IEEE/ACM Transactions on Networking* 18, 6 (2010), 1949 – 1962.

[20] Xi Chen, Prateek Gangwal, and Daji Qiao. 2012. RAM: Rate Adaptation in Mobile Environments. *IEEE Transactions on Mobile Computing* 11, 3 (2012), 464–477.

[21] Lara Deek, Eduard Garcia-Villegas, Elizabeth Belding, Sung-Ju Lee, and Kevin Almeroth. 2013. Joint Rate and Channel Width Adaptation for 802.11 MIMO Wireless Networks. In *Proc. of IEEE SECON*.

[22] Felix Fietkau. 2010. Minstrel_ht: new rate control module for 802.11n. https://lwn.net/Articles/376765/

[23] Yasaman Ghasempour, Muhammad Kumail Haider, Carlos Cordeiro, Dimitrios Koutsonikolas, and Edward Knightly. 2018. Multi-Stream Beam-Training for mmWave MIMO Networks. In *Proc. of ACM MobiCom*.

[24] Muhammad Kumail Haider and Edward W. Knightly. 2016. Mobility Resilience and Overhead Constrained Adaptation in Directional 60 GHz WLANs: Protocol Design and System Implementation. In *Proc. of ACM MobiHoc*.

[25] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2010. Predictable 802.11 packet delivery from wireless channel measurements. In *Proc. of ACM SIGCOMM*.

[26] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *ACM SIGCOMM CCR* 41, 1 (2011), 53.

[27] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. 2011. Augmenting data center networks with multi-gigabit wireless links. In *Proc. of ACM SIGCOMM*.

[28] Haitham Hassanieh, Omid Abari, Michael Rodriguez, Mohammed Abdelghany, Dina Katabi, and Piotr Indyk. 2018. Fast Millimeter Wave Beam Alignment. In *Proc. of ACM SIGCOMM*.

[29] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. 2001. Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proc. of ACM MobiCom*.

[30] Jingqi Huang, Song Wang, and Xinyu Zhang. 2020. Demystifying Millimeter-Wave V2X: Towards Robust and Efficient Directional Connectivity Under High Mobility. In *Proc. of ACM MobiCom*.

[31] Sooyoung Hur, Taejoon Kim, David J Love, James V Krogmeier, Timothy A Thomas, and Amitava Ghosh. 2013. Millimeter Wave Beamforming for Wireless Backhaul and Access in Small Cell Networks. *IEEE Transactions on Communications* 61, 10 (2013), 4391âĂŞ4403.

[32] IEEE 802.11 Working Group. 2012. IEEE 802.11ad, Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band. (2012).

[33] IEEE 802.11 Working Group. 2017. Enhanced throughput for operation in license-exempt bands above 45 GHz. *IEEE P802.11ay/D0.3* (2017).

[34] National Instruments. 2017. Introduction to the NI mmWave Transceiver System Hardware - National Instruments. http://www.ni.com/white-paper/53095/en/

[35] Glenn Judd, Xiaohui Wang, and Peter Steenkiste. 2008. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *Proc. of ACM MobiSys*.

[36] Ad Kamerman and Leo Monteban. 1997. WaveLAN II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal* (1997).

[37] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. 2017. IEEE 802.11ac Link Adaptation Under Mobility. In *Proc. of IEEE LCN*.

[38] Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and Daji Qiao. 2006. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *Proc. of IEEE INFOCOM*.

[39] Lito Kriara and Mahesh Marina. 2015. SampleLite: A hybrid approach to 802.11n link adaptation. *ACM SIGCOMM Computer Communication Review* 45, 2 (2015), 4–13.

[40] Ernest Kurniawan, Lin Zhiwei, and Sumei Sun. 2017. Machine Learning-Based Channel Classification and Its Application to IEEE 802.11ad Communications. In *Proc. of IEEE GLOBECOM*.

[41] Zeqi Lai, Y. Charlie Hu, Yong Cui, Linhui Sun, and Ningwei Dai. 2017. Furion: Engineering High-Quality Immersive Virtual Reality on Today's Mobile Devices. In *Proc. of ACM MobiCom*.

[42] Chi-Yu Li, Chunyi Peng, Songwu Lu, and Xinbing Wang. 2012. Energy-based Rate Adaptation for 802.11N. In *Proc. of ACM MobiCom*.

[43] Zhinus Marzi, Dinesh Ramasamy, and Upamanyu Madhow. 2016. Compressive Channel Estimation and Tracking for Large Arrays in mmWave Picocells. *IEEE Journal of Selected Topics in Signal Processing* 10, 3 (2016), 514âĂŞ527.

[44] Jiayi Meng, Sibendu Paul, and Y. Charlie Hu. 2020. COTERIE: Exploiting Frame Similarity to Enable High-Quality Multiplayer VR on Commodity Mobile Devices. In *Proc. of ACM ASPLOS*.

[45] D. Nguyen and J. J. Garcia-Luna-Aceves. 2011. A practical approach to rate adaptation for multi-antenna systems. In *Proc. of IEEE ICNP*.

[46] Thomas Nitsche, Guillermo Bielsa, Irene Tejado, Adrian Loch, and Joerg Widmer. 2015. Boon and Bane of 60 GHz Networks: Practical Insights into Beamforming, Interference, and Frame Level Operation. In *Proc. of ACM CoNEXT*.

[47] Ioannis Pefkianakis, Yun Hu, Starsky H.Y. Wong, Hao Yang, and Songwu Lu. 2010. MIMO Rate Adaptation in 802.11N Wireless Networks. In *Proc. of ACM MobiCom*.

[48] Swetank Kumar Saha, Shivang Aggarwal, Rohan Pathak, Dimitrios Koutsonikolas, and Joerg Widmer. 2019. MuSher: An Agile Multipath-TCP Scheduler for Dual-Band 802.11ad/ac Wireless LANs. In *Proc. of ACM MobiCom*.

[49] S. K. Saha, H. Assasa, A. Loch, N. M. Prakash, R. Shyamsunder, S. Aggarwal, D. Steinmetzer, D. Koutsonikolas, J. Widmer, and M. Hollick. 2018. Fast and Infuriating: Performance and Pitfalls of 60 GHz WLANs Based on Consumer-Grade Hardware. In *Proc. of IEEE SECON*.

[50] Swetank Kumar Saha, Yasaman Ghasempour, Muhammad Kumail Haider, Tariq Siddiqui, Paulo De Melo, Neerad Somanchi, Luke Zakrajsek, Arjun Singh, Roshan Shyamsunder, Owen Torres, Daniel Uvaydov, Josep Miquel Jornet, Edward Knightly, Dimitrios Koutsonikolas, Dimitris Pados, Zhi Sun, and Ngwe Thawdar. 2019. X60: A Programmable Testbed for Wideband 60 GHz WLANs with Phased Arrays. *Computer Communications* (2019).

[51] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. 2010. AccuRate: Constellation Based Rate Estimation in Wireless Networks. In *Proc. of USENIX NSDI*.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 56 (2014), 1929–1958.

[53] Daniel Steinmetzer, Daniel Wegemer, and Matthias Hollick. 2017. Talon Tools: The Framework for Practical IEEE 802.11ad Research. https://seemoo.de/talon-tools

[54] Daniel Steinmetzer, Daniel Wegemer, Matthias Schulz, Joerg Widmer, and Matthias Hollick. 2017. Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11 ad Devices. In *Proc. of the 11th ACM CoNEXT*.

[55] Li Sun, Souvik Sen, and Dimitrios Koutsonikolas. 2014. Bringing Mobility-Awareness to WLANs using PHY layer information. In *Proceedings of the ACM CoNEXT*.

[56] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parmesh Ramanathan. 2015. 60 GHz Indoor Networking Through Flexible Beams: A Link-Level Profiling. In *Proc. of ACM SIGMETRICS*.

[57] Sanjib Sur, Xinyu Zhang, Parmesh Ramanathan, and Ranveer Chandra. 2016. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *Proc. of USENIX NSDI*.

[58] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. 2009. Cross-Layer Wireless Bit Rate Adaptation. In *Proc. of ACM SIGCOMM.*
[59] Yuyang Wang, Murali Narasimha, and Robert W. Heath. 2018. MmWave Beam Prediction with Situational Awareness: A Machine Learning Approach. In *Proc. of IEEE SPAWC.*
[60] Teng Wei and Xinyu Zhang. 2017. Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support. In *Proc. of ACM MobiCom.*
[61] Teng Wei, Anfu Zhou, and Xinyu Zhang. 2017. Facilitating Robust 60 GHz Network Deployment By Sensing Ambient Reflectors. In *Proc. of USENIX NSDI.*
[62] Michael S. Y. Wong, Jeffrey M. Gilbert, and Craig H. Barratt. 2008. Wireless LAN using RSSI and BER parameters for transmission rate adaptation. US Patent 7,369,510.
[63] Starsky H. Y. Wong, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. 2006. Robust Rate Adaptation for 802.11 Wireless Networks. In *Proc. of ACM MobiCom.*
[64] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise Power Delay Profiling with Commodity WiFi. In *Proc. of ACM MobiCom.*
[65] Masoud Zarifneshat, Li Xiao, and Jiliang Tang. 2019. Learning-based Blockage Prediction for Robust Links in Dynamic Millimeter Wave Networks. In *Proc. of IEEE SECON.*
[66] Ding Zhang, Panneer Selvam Santhalingam, Parth Pathak, and Zizhan Zheng. 2019. Characterizing Interference Mitigation Techniques in Dense 60 GHz mmWave WLANs. In *Proc. of ICCCN.*
[67] Jiansong Zhang, Kun Tan, Jun Zhao, Haitao Wu, and Yongguang Zhang. 2008. A Practical SNR-Guided Rate Adaptation. In *Proc. of IEEE INFOCOM.*
[68] 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 ACM MobiCom.*
[69] Anfu Zhou, Leilei Wu, Shaoqing Xu, Huadong Ma, Teng Wei, and Xinyu Zhang. 2018. Following the Shadow: Agile 3-D Beam-Steering for 60 GHz Wireless Networks. In *Proc. of IEEE INFOCOM.*
[70] A. Zhou, X. Zhang, and H. Ma. 2017. Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering. In *Proc. of IEEE INFOCOM.*
[71] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. 2012. Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *Proc. of ACM SIGCOMM.*
[72] Yibo Zhu, Xia Zhou, Zengbin Zhang, Lin Zhou, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. 2014. Cutting the Cord: a Robust Wireless Facilities Network for Data Centers. In *Proc. of ACM MobiCom.*

**Algorithm 1** LiBRA

```
N, M ← Number of Tx, Rx beams
minMCS, maxMCS ← Minimum, maximum supported MCS
minTput, minCDR ← Throughput, CDR thresholds for working MCS
T ← T_o, k ← 0

function BA()
    maxSNR ← −∞
    for beamPairID ← 0 to (N x M) − 1 do
        SNR ← measure_snr(beamPairID)
        if SNR > maxSNR then
            bestBeamPairID ← beamPairID
            maxSNR ← SNR
    curr_beam_pair ← bestBeamPairID

function RA(initMCS, initTput)
    maxTput ← initTput
    for MCS ← initMCS to minMCS step − 1 do
        tput ← measureTput(MCS)
        if tput < maxTput then
            if isWorking(MCS + 1) then
                curr_mcs ← MCS + 1
            else
                BA()
                RA(initMCS, 0)
            return
        maxTput ← tput
    if isWorking(MCSmin) then
        curr_mcs ← MCSmin
    else
        BA()
        RA(initMCS, 0)

function isWorking(MCS)
    return measureTput(MCS) ≥ minTput && CDR(MCS) ≥ minCDR

function selectAction(frameID)
    if probeMCS then
        if curr_tput < prev_tput or noACK then
            k ← k + 1
            curr_mcs ← curr_mcs − 1
        else
            k ← 0
            if frameID % 2 == 1 then
                frameID ← frameID + 1
        T ← T_o · min(2^k, 2^5)
        probeMCS ← false
        return
    T ← T − 1
    if noACK then
        if curr_mcs < 6 or BAOverhead < BAOverheadThreshold then
            BA()
            RA(curr_mcs, 0)
        else
            RA(curr_mcs − 1, 0)
    else
        if frameID % 2 == 0 then
            return
        metrics ← updateMetrics(frameID, frameID − 1)
        action ← classifyBaRaNa(metrics, prev_metrics)
        if action == BA then
            BA()
            RA(curr_mcs, curr_tput)
            T ← T_o
        else if action == RA then
            RA(curr_mcs − 1, curr_tput)
            T ← T_o
        prev_metrics ← metrics

while true do
    sendFrame(frameID)
    selectAction(frameID)
    CDR(curr_mcs) ← updateCDR()
    if T == 0 and CDR(curr_mcs) > CDR_ORI(curr_mcs) then
        curr_mcs ← curr_mcs + 1
        probe_mcs ← true
        prev_tput ← curr_tput
    frameID ← frameID + 1
```

# A APPENDIX

## A.1 *LiBRA* Algorithm

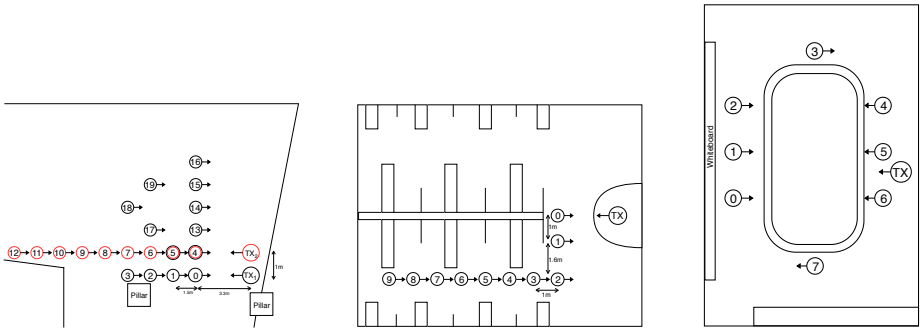
*LiBRA* is described in Algorithm 1.

## A.2 Dataset collection: Measurement environments and scenarios

In this section, we provide details about the environments we used to collect the main dataset (Table 1) and the various scenarios we considered.

*A.2.1 Environments.* We collected a dataset by taking measurements in multiple environments within a campus building.
**Lobby.** This is a large open space with glass panels covering the upper part and metallic sheets covering the lower part of one side and a wall on the other side. It is shown in Fig. 14a along with the various Tx and Rx positions we used for our measurements.
**Lab.** This is an $11.8 \times 9.2 \times 3.4$ m$^3$ space with 4 rows of desks surrounded by metallic storage cabinets and white boards (Fig. 14b).
**Conference Room.** This is a $10.4 \times 6.8 \times 3.2$ m$^3$ space with a large white board covering one of the walls (Fig. 14c). There are metallic cabinets, a large desk in the center of the room, and many chairs.
**Corridors.** We performed measurements in 3 corridors of width 1.74 m, 3.2 m, and 6.2 m. In the lobby, the conference room, and the corridors, the Tx and Rx antennas were kept at a height of 1.4 m. In the lab, LOS at this height was blocked by furniture, hence we placed the Tx antenna at a height of 2.05 m and the Rx antenna at a height of 1.25 m.

*A.2.2 Linear and/or angular displacement scenario – details.* Here we provide details about the measurements involved in the displacement scenario in §4.2.
**Lobby:** We first fixed the Tx at position $Tx_1$ and the Rx at position 0 (Fig. 14a). We then moved the Rx while keeping the orientation fixed along three directions – backwards, laterally, and diagonally – and took measurements at multiple positions in each direction. We also took measurements by rotating the Rx at positions 2 and 19. We then took a second set of measurements with the Tx fixed at position $Tx_2$ and the Rx at 9 positions.
**Lab:** The Tx was fixed and the Rx was placed at 10 different positions, starting at position 0. We also took measurements by rotating the Rx at positions 2, 5, and 8.
**Conference room:** The Tx was fixed and the Rx was placed at different positions and orientations around the table, shown in Fig. 14c, starting at position 0. Note that for positions 4, 5, 6 and 7, the Rx is facing in the same direction as the Tx and communication is enabled through reflections. We also performed rotations at positions 0 and 4.
**Corridors:** In the narrow corridor, we performed measurements at 17 different Rx positions starting from a distance of 2.5 m away from the Tx and moving back in steps of 1.25 m, with the Tx and Rx always facing each other. In the 2 wider corridors, we fixed the Tx and performed measurements at 10 different Rx positions at steps of 1.25 m with the Tx and Rx facing each other at all times. We also performed rotations 5 m, 10 m, and 15 m away from the Tx.




(a) Lobby. (b) Lab. (c) Conference Room.

**Figure 14: Environments and measurement positions. The arrows show the Tx and initial Rx orientations.**