WiFi-Assisted 60 GHz Wireless Networks

Sanjib Sur[∞,*], Ioannis Pefkianakis[¬], Xinyu Zhang[▷], Kyu-Han Kim[¬]
University of Wisconsin-Madison[∞], Hewlett Packard Labs[¬], University of California San Diego[▷]
sur2@wisc.edu, ioannis.pefkianakis@hpe.com, xiz368@eng.ucsd.edu, kyu-han.kim@hpe.com

ABSTRACT

Despite years of innovative research and development, gigabit-speed 60 GHz wireless networks are still not mainstream. The main concern for network operators and vendors is the unfavorable propagation characteristics due to short wavelength and high directionality, which renders the 60 GHz links highly vulnerable to blockage and mobility. However, the advent of multi-band chipsets opens the possibility of leveraging the more robust WiFi technology to assist 60 GHz in order to provide seamless, Gbps connectivity. In this paper, we design and implement *MUST*, an IEEE 802.11-compliant system that provides seamless, high-speed connectivity over multi-band 60 GHz and WiFi devices. *MUST* has two key design components: (1) a WiFi-assisted 60 GHz link adaptation algorithm, which can instantaneously predict the best beam and PHY rate setting, with zero probing overhead; and (2) a proactive blockage detection and switching algorithm which can re-direct ongoing user traffic to the robust interface within sub-10 ms latency. Our experiments with off-the-shelf 802.11 hardware show that *MUST* can achieve 25-60% throughput gain over state-of-the-art solutions, while bringing almost 2 orders of magnitude cross-band switching latency improvement.

CCS CONCEPTS

• **Networks** → **Wireless access networks**; **Network protocol design**;

KEYWORDS

60 GHz; Millimeter-Wave; IEEE 802.11ad; Session Transfer

1 INTRODUCTION

The multi-GHz unlicensed spectrum at the 60 GHz millimeter-wave frequency band promises to shift current WiFi-experience from "Wireless" to Gbps "Wire-like". With up to 14 GHz of free spectrum [1], 60 GHz offers a foundation for next-generation bandwidth-intensive applications, such as uncompressed video streaming, snap wireless file synchronization, wireless virtual and augmented reality, wireless data centers, and Gbps Internet access. Multiple standardization efforts such as IEEE 802.11ad [24], 802.15.3c [12], ECMA [20] and millimeter-wave products [48] are tailored to support such applications, and promise upto 7 Gbps of wireless bit-rate. These advances and research demonstrations [23, 54, 63] have also led millimeter-wave technology to be recommended as a key enabler for multi-Gbps 5G cellular networks [1–3].

However, the short wavelength (5 mm at 60 GHz) and weak reflection characteristics (compared to LTE/WiFi signals) render 60 GHz links highly vulnerable to channel propagation loss [51, 60]. In the same environment, a 60 GHz link suffers from around 1000× higher signal strength loss compared to typical LTE/WiFi [53]. Millimeter-wave devices overcome such challenge by focusing RF energy towards narrow spatial direction through beamforming using phased-array antennas. Maintaining beams from access point (AP)/base station toward mobile users is a challenging task and 60 GHz links formed via such narrow beams get highly affected during mobility and human blockage [15, 47, 53, 54].

Our experiments with off-the-shelf IEEE 802.11ad platform in indoor settings show the potential of incorporating 60 GHz links in enterprise networks, but also expose its limitations. While a few of the observations are broadly expected, our measurement study quantifies the benefit and limitations of adopting 60 GHz in today's enterprises. In static, Line-Of-Sight (LOS) settings, 60 GHz can achieve 2.5 Gbps throughput, which is 4× higher than state-of-the-art WiFi (IEEE 802.11ac)[1]. However, during blockage and mobility where WiFi can transmit at full speed, 60 GHz links may occasionally get fully disconnected for several hundred milliseconds. Although IEEE 802.11ad transmitter and receiver can recover from such dynamics by searching for the best signal-strength beam at runtime, our experiments show that the search algorithm itself takes hundreds of ms to converge. Recent research proposals [6, 35, 47, 54] seek to minimize the convergence time to the best beam. However, such approaches either assume quasi-stationary 60 GHz links, or have limitations to find the settings at low cost, in dynamic scenarios with today's commodity devices. Even if an "Oracle" could instantaneously find the best beam, our results show multiple occasions where 60 GHz link establishment is not feasible, due to the fundamental physics of the millimeter-wave communication. Deploying denser APs [58, 64] or reflectors [5] could improve but does not guarantee robust 60 GHz connectivity despite the high deployment cost.

Fortunately, the advent of multi-band WiFi chipsets [59] opens the possibility of leveraging the more robust WiFi interface in those scenarios where 60 GHz fundamentally performs poorly, to enable robust connectivity. Specifically, WiFi's omni-directional view of the wireless channel could be leveraged to design highly-adaptive 60 GHz beam and PHY rate selection. In the scenarios where a 60 GHz link cannot be established due to obstacles, WiFi could provide admittedly lower speeds, but still a robust communication anchor.

Designing WiFi-assisted millimeter-wave communication is a challenging problem for *three* key reasons. *First*, while omni-directional WiFi Channel State Information (CSI) could provide a hint to the

[*] Sanjib Sur interned at Hewlett Packard Labs, Palo Alto during this work.

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

ACM ISBN 978-1-4503-4916-1/17/10…$15.00
https://doi.org/10.1145/3117811.3117817

[1] In this paper, we use 60 GHz or IEEE 802.11ad, and WiFi or IEEE 802.11ac interchangeably.

most dominant propagation path (and hence to the best 60 GHz beam), it is coarse grained estimation, and often leads to erroneous beam selection, as shown by our experiments. *Second*, simultaneously transmitting data over multiple interfaces with highly heterogeneous speeds (*i.e.* 60 GHz and WiFi) degrades application performance over TCP, compared to a single-interface transmission. To our surprise, we have observed that TCP throughput can drop by up to 5×, when both 60 GHz and WiFi are used simultaneously. *Finally*, designing seamless 60 GHz-WiFi interface switching, which meets the sub-10 ms latency requirements of many real-time applications, remains an open problem. Existing off-the-shelf multi-band devices adopt a reactive interface switching approach, which requires multiple seconds to identify and switch to the best interface.

In this paper, we present *MUST* (*MU*lti-band fast *S*ession *T*ransfer), an IEEE 802.11-compliant system which provides high-speed, robust connectivity over 60 GHz and WiFi multi-band devices, in dynamic indoor environments. *MUST* achieves its objective by introducing *two* key design components: (1) *Fast link adaptation:* A WiFi-assisted 60 GHz link adaptation module that can instantaneously predict the best 60 GHz beam and PHY rate setting, with zero 60 GHz probing, under blockage and mobility; (2) *Seamless switching:* A low-latency (sub-10 ms), proactive switching algorithm to reroute traffic from highly fragile 60 GHz links to WiFi, without breaking the existing connection. *MUST* design is built upon *two* key observations: (1) Human mobility and blockage affect the dominating path of the 60 GHz channel and thus best beam, at much slower pace (order of 100s of ms). While a small scale variation can still affect channel quality of the best beam, it does not necessarily change the entire best beam direction in a practical system. Thus, instead of relying on the existing method of link adaptation that itself can take hundreds of ms to converge, *MUST* uses out-of-band information with appropriate modeling that provides the hint for the 60 GHz channel's most dominating path and best link settings. (2) When LOS is blocked, 60 GHz suffers from highly dynamic, occasional zero-throughput connectivity that eventually degrades upper layer performance. In order to achieve smooth real-time response, it is far better to proactively move to WiFi interface to survive blockage.

We implement *MUST* in an off-the-shelf IEEE 802.11ad/ac multi-band AP platform without any modifications to the end-user device or additional standards support. Different from existing efforts, which focus on a single aspect of 60 GHz communication (*e.g.* link adaptation [6, 35, 47, 54]), *MUST* is a full-fledged system design which introduces optimizations across the protocol stack, considering system-level constraints and overheads. Our experiments show that *MUST* can accurately approximate the best performance link settings under various channel dynamics in enterprise environment. Even after including all out-of-band probing and interface switching related overheads, it stays within 10% of the optimum and achieves 25-60% throughput gain over state-of-the-art solutions. While proactively switching to WiFi can occasionally incur performance loss (less than 6% on average), *MUST* improves the switching latency by almost 2 orders of magnitude.

2 BACKGROUND

2.1 IEEE 802.11ad and 802.11ac

60 GHz: IEEE 802.11ad [24] devices operate on the 60 GHz unlicensed spectrum, which uses 2.16 GHz of channel bandwidth per link, with peak PHY rate upto 7 Gbps (the commodity IEEE 802.11ad devices in our experiments support upto 4.62 Gbps). IEEE 802.11ad devices can overcome the high channel propagation loss of 60 GHz signal, by using multi-antenna phased-array to steer RF energy towards narrow spatial direction (*i.e.*, beamforming). Specifically, they use analog amplitude and phase shifters that are configured according to a predefined codebook of beamforming coefficients, with each codebook entry generating a *sector* (or beam). Due to the small form factor of 60 GHz RF components and antennas, large phased-arrays can be integrated into mobile devices.

IEEE 802.11ad supports a beamforming training (BFT) process to discover the highest signal strength Tx and Rx beams between a pair of devices. BFT comprises of a mandatory Sector Level Sweep (SLS) phase, and an optional Beam Refinement Phase (BRP), which hierarchically evaluate the Tx and Rx beam combinations, to identify the best one [24, 53].

WiFi: IEEE 802.11ac [25] devices operate on the 5 GHz band, with up to 160 MHz of channel bandwidth. Although the highest 802.11ac PHY rate is 6.9 Gbps, the rates which are used by typical off-the-shelf WiFi devices are up to 866.7 Mbps. This is because mobile user devices do not support more than 2-stream MIMO due to energy and antenna size constraints. Moreover, 160 MHz channels are often disabled by AP vendors in enterprise and campus settings due to mutual interferences with legacy 802.11a/n devices.

2.2 Fast Session Transfer

To enable seamless transition to IEEE 802.11ac interface on the same device, IEEE 802.11ad supports an optional Fast Session Transfer (FST) feature [24]. Specifically, IEEE 802.11ad specifies the MAC-level control and coordination procedures between the AP and user, which allows traffic to migrate in between the WiFi and 60 GHz interfaces, transparent to higher layer protocols.

3 MEASUREMENTS ON OFF-THE-SHELF HARDWARE

In this section, we conduct experiments with off-the-shelf hardware to understand how 60 GHz compares with WiFi in various indoor settings. While few of the measurement observations are broadly expected, our goal here is to assess the benefits and limitations of incorporating 60 GHz in typical enterprise settings. The findings of this section form the basis of our arguments behind WiFi-assisted 60 GHz wireless networks.

3.1 Experimental Platform and Methodology

Platform. We conduct all our experiments using commodity APs and end-user devices (Figure 1(b)) that integrate both 60 GHz and WiFi radios on the same device. The 60 GHz radio at AP consists of a baseband chipset that supports 2.16 GHz channel bandwidth and up to 16-QAM modulation level, with 4.62 Gbps peak PHY rate. In addition, a separate RFIC controls an integrated 32-element phased-array antenna (arranged in a 4×8 matrix), that can generate directional beams in three-dimensional space. The spatial direction

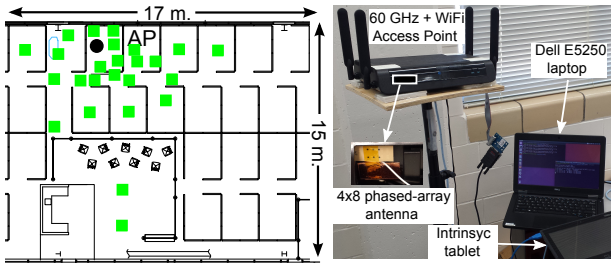

Figure 1: (a) **Experimental floorplan. Square boxes represent static users' position.** (b) **Experimental platform.**

of the beams depends on the phased-array antenna configurations, and have been optimized by the vendor to minimize the overlap, and hence to maximize spatial coverage. Several example transmit beam directions in our platform are shown in Figure $5(a)$, where each dot represents the strongest direction of the beam. The azimuth and elevation angles of the beams can span within {-87.5°, 90°} and {-45°, 45°} respectively[2]. Phased-array beamforming training and rate adaptation algorithms are implemented at the firmware of the baseband chipset, which sends the control signals to the RFIC, to steer the beam directions.

Our AP platform's WiFi interface supports 3 antennas, upto 80 MHz bandwidth and MU-MIMO technology. Its peak PHY rate is 866.7 Mbps. Both the 60 GHz and WiFi radios are connected to a general-purpose System-on-Chip (*a.k.a.* host) via PCIe bus, which uses a dual-core 1.7 GHz CPU with a 512 MB DDR3 memory, and runs OpenWrt [4]. The host system implements legacy functionalities including processing and forwarding packets from backhaul Ethernet to the radios. An off-the-shelf platform with similar hardware-software architecture is available here [30]. At the user side, we use Dell laptops [17] and Intrinsyc tablets [27] that also integrate the 60 GHz and WiFi chipsets on the same device. User devices have the same 60 GHz radio as the AP, but their WiFi interfaces only have 2 antennas.

Methodology. For our experiments, we modified the kernel and firmware of our AP, to periodically collect fine-grained statistics such as throughput, PHY rate, beam directions, for each connected user device. Using our testbed, we conduct experiments in an enterprise and a campus building.

3.2 Performance in Multiple Settings

We start our measurement study by comparing 60 GHz and WiFi throughput in the building floor shown in Figure $1(a)$. For our experiments, we generate saturated TCP traffic from an AP to a user, in controlled settings, without external channel interferences, or perturbations in the physical environment.

Static settings. We place the AP at two different heights from the floor (2.6 ft table and 9 ft tripod), and a user at 25 different spots (represented by rectangles) in Figure $1(a)$. Figure $2(a)$ shows the distribution of the 60 GHz and WiFi throughput difference ($Thr_{60} - Thr_{WiFi}$) for the two AP placements. While 60 GHz can achieve up to 1.7 Gbps higher throughput than today's state-of-the-art WiFi, it is typically better in LOS settings. Overall, 60 GHz outperforms WiFi in more settings when the AP is placed higher, because more non-blocking links can be established. When an obstacle blocks the LOS, WiFi can achieve up to 405 Mbps higher throughput (Figure $2(a)$). In most of the settings where WiFi performs better, 60 GHz link achieves almost zero throughput, or it gets disconnected. Figure $2(c)$ further shows that in open LOS, 60 GHz can achieve more than 1.5 Gbps throughput even beyond 10 m from the AP (typical AP separation distance in today's enterprise). However, obstacle blockage may disconnect 60 GHz links, and even beam searching and steering may not be able to establish any connection in Non-Line-Of-Sight (NLOS). WiFi on the other hand provides more stable and robust connection in NLOS.

Spatial and temporal dynamics. 60 GHz channel is more dynamic than WiFi in both space and time, because of its higher wireless frequency [45]. The dynamic performance of 60 GHz channel in various static and pedestrian mobility settings is shown in the boxplot throughput distribution of Figure $2(b)$ within the experimental area in Figure $1(a)$. While 60 GHz throughput can reach upto 2.2 Gbps in the aforementioned settings, more than 25% of the links show no connection. WiFi links never disconnect and the throughput ranges from 80 to 480 Mbps.

Summary. 60 GHz communication shows high potential to achieve multi-Gbps speed with today's off-the-shelf IEEE 802.11ad hardware, which can outperform state-of-the-art WiFi by more than 1 Gbps in LOS. However, WiFi is robust under blockage. *Thus, 60 GHz and WiFi can be complimentary to each other in a typical indoor setting.*

3.3 Convergence to the Best-Speed Setting

The 60 GHz poor performance instances shown by our experiments are not only attributed to channel dynamics (*e.g.*, blockage), but also to the slow convergence to the best-throughput beam and rate settings [53, 64]. We conduct controlled experiments to quantify such overhead. Specifically, a static LOS user device is placed 50 cm away from the AP, and a human body intentionally blocks in between. We monitor the SNR change of the current transmit beam at every 1 ms interval and measure the time elapsed between removing the blockage and returning to the best beam and rate (that is used before blockage occurrence). Figure $2(d)$ shows the distribution of 60 GHz link recovery time for 50 blockage scenarios. We observe very high convergence times, varying from 10 ms to 910 ms, with a median of 295 ms.

The reason behind such long convergence time is that, existing design lacks effective mechanisms to determine when to trigger beam searching, which is left as an open option in IEEE 802.11ad. For example, our platform collects the Packet-Error-Rate (PER) statistics over multiple probe packets to evaluate the reliability of a beam [40]. It will trigger a beam search only if the beam's "reliability" drops below a threshold, which leads to poor responsiveness. Although recent research proposals hold potential for fast convergence, they either assume quasi-stationary 60 GHz links [54] or require constant customized probing [32, 47]. After triggering the beam searching process, beam scan for small-sized phased-array (32 elements) is a fast operation. Our AP's IEEE 802.11ad radio can perform SLS and BRP scans in 1.25 ms. However, for larger phased-array platforms (*e.g.* 1000-elements), the IEEE 802.11ad SLS and BRP procedure can still be on the order of seconds [53, 64].

[2]Note that, range of the elevation angle is typically smaller due to restricted vertical movement of the user devices.

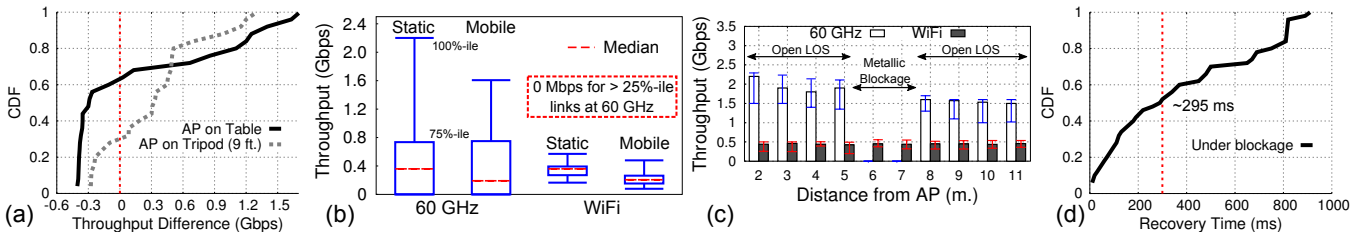

Figure 2: (a) Throughput difference between 60 GHz and WiFi links for two AP heights across 25 static locations. (b) Spatial and temporal dynamics. (c) 60 GHz and WiFi throughput in open LOS and NLOS. (d) Distribution of 60 GHz static link recovery time across 50 LOS blockage scenarios.

Summary: Probing-based 60 GHz beam and rate adaptation algorithms result in long convergence time in dynamic settings, even for a small-sized phased-array. *A probe-less fast link adaptation approach is desired to utilize the 60 GHz channel capacity, under channel dynamics.*

3.4 Joint 60 GHz and WiFi Operation

3.4.1 Enabling Single vs. Multiple Interfaces

Our experimental results corroborate the use of both 60 GHz and WiFi to achieve high-speed and robust connectivity. Indeed, existing schemes [29, 38, 41–43] have already considered simultaneously leveraging all the network interfaces (*e.g.* LTE and WiFi) to boost performance. We seek to understand if such approaches can work in our 60 GHz and WiFi settings. We conduct controlled experiments in static, open LOS, where the AP generates TCP traffic to one user device in two scenarios: (1) AP and user communicate only through 60 GHz, and (2) AP enables parallel transmissions over both 60 GHz and WiFi. In the latter scenario, our AP uses the open source Linux bonding driver [16] to bond 60 GHz and WiFi, and expose a single virtual IP/MAC address towards the network stack. We modified the bonding driver to push packets in parallel at both interfaces. Figure 3(a) shows the average (min/max) throughput for both scenarios, and for various 60 GHz link performance settings, as indicated by the 60 GHz PHY rates[3]. To our surprise, we observe that the TCP throughput drops by up to 5× when both interfaces are used in parallel! While the 60 GHz interface alone can achieve more than 2.5 Gbps, parallel 60 GHz & WiFi transmissions barely achieve 500 Mbps.

Root cause. The above experimental result is surprising to say the least. The root cause turned out to be hidden inside the TCP's congestion control. Specifically, due to highly heterogeneous speed links, we observe almost 50% of the TCP packets to arrive at the receiver out-of-order. Out-of-order packets result in duplicate ACKs and trigger the *Fast Retransmit* at the TCP sender [14], which results in unnecessary retransmissions. Further, limited receiver buffer space causes incomplete TCP packets that arrived early from the faster 60 GHz interface, to be frequently dropped due to buffer overflow. Both issues cause the sender to reduce the TCP congestion window (cwnd). Figure 3(b) shows the evolution of cwnd over time during a 1 GB downlink TCP transfer, for both single and parallel transmissions[4]. When both interfaces are used, the cwnd is severely affected and always remains smaller than 250 packets. However, under the same setup when packets are sent solely through 60 GHz, the cwnd quickly converges to approximately 2900 packets and consumes almost 4× less time to finish the transfer.

[3]Rates 1251 and 2502 Mbps are hardware-disabled in our platform.
[4]Our AP implements TCP CUBIC [26], default option for Linux.

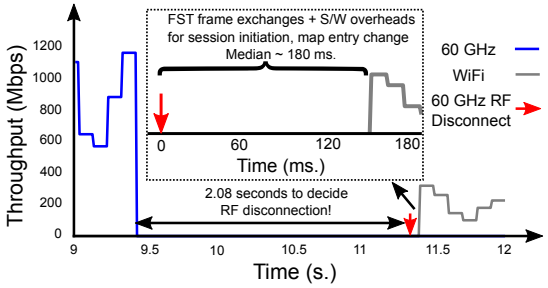

Figure 4: Temporal analysis of our platform's FST.

Does MPTCP help? At first blush, it may seem that the problem can be solved by enabling flow control over both paths using multipath TCP (MPTCP) [18]. Unfortunately, independent studies [21, 43] have shown that MPTCP performs poorly over heterogeneous paths, even for stable Gbps data-center links. This is because out-of-order TCP packets and limited receiver buffer size similarly affect MPTCP's performance. Our experiments also verify that using the 60 GHz interface alone achieves 7% to 45% higher throughput compared to MPTCP over 60 GHz and WiFi. The authors in [43] proposed extensive modifications to MPTCP for heterogeneous-speed paths, by opportunistically retransmitting TCP sequences over the highest-speed path, or by reducing congestion window over slow paths. However, both approaches waste network resources. Allocating larger buffer size at the receiver may reduce buffer overflow, but it does not prevent out-of-order packet reception. Besides, large receiver buffer size severely affects TCP's response to real-time mobile applications such as VR video streaming.

Summary: *Simultaneously transmitting through the 60 GHz and WiFi interfaces can severely affect TCP performance. Therefore, we opt to select one interface at a time, to accommodate individual user's traffic and switch to the best interface without breaking the existing connection.* We next evaluate the cost associated with such interface switching.

3.4.2 Interface Switching Overheads

IEEE 802.11ad FST overheads. Our platform implements the IEEE 802.11ad FST feature (Section 2.2, [24]), to allow for IEEE 802.11-compliant interface switching. The latency related to FST control

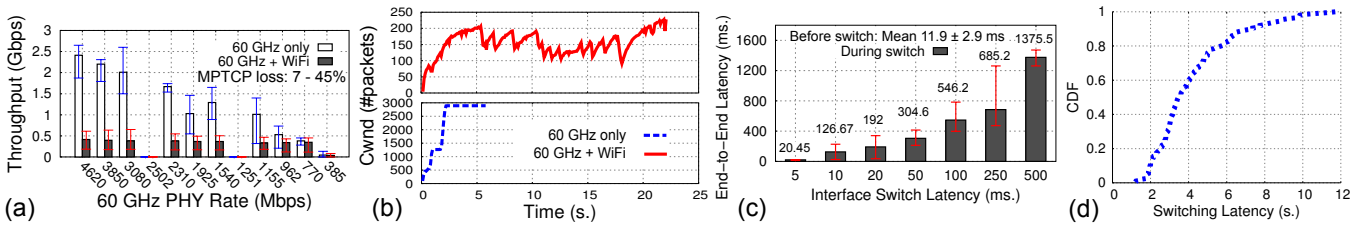

Figure 3: TCP's behavior under parallel transmission: (a) Throughput under various 60 GHz PHY rates. (b) Congestion window evolution for 1 GB downlink transmission. (c) TCP end-to-end latency under varying interface switch latency. (d) Distribution of switching latency in reactive FST.

packet and coordination handshake overhead at the radio is typically less than 2 ms. The implemented FST manager is a user level daemon, which runs on top of the kernel's bonding driver and triggers interface switching upon 60 GHz link disconnection. Our measurements show that the time elapsed between FST trigger and completion is on median 180 ms in our platform. This high latency is specific to the off-the-shelf platform and mainly attributed to the overheads of initiating a new session, various OS and firmware level message exchanges and under-optimized software implementation.

Impact of overheads on TCP. Unfortunately, such interface switching cost can severely degrade TCP performance. We experimentally evaluate the impact, by modifying our kernel's bonding driver to artificially add controlled delays, when switching between interfaces. Figure 3(c) shows how TCP's end-to-end latency is affected by various interface switching latencies. While the increase of the TCP's end-to-end latency after introducing 5 ms switching latency is almost negligible, we observe that TCP performs poorly when the switching latency goes beyond 10 ms. For example, a 100 ms switching latency can exacerbate end-to-end latency by more than 500 ms. This is because all the TCP packets are dropped during interface switching, resulting in packet retransmissions and congestion window shrinkage.

Limitations of reactive FST. Besides minimizing interface switching costs, an important design decision is when to trigger FST. The off-the-shelf platform takes a reactive approach and switches to WiFi, only after 60 GHz link gets disconnected based on a link timeout. For example, Figure 4 shows the switching procedure of our platform's FST, in a scenario where 60 GHz link breaks, before the 9.5^{th} second. It takes approximately 2.26 seconds for the FST to switch to WiFi. Specifically, the 60 GHz firmware and RF front-end needs 2.08 seconds to decide the link disconnection, and send the disconnection event to the FST daemon. Upon disconnection, the switching cost is approximately 180 ms. Our experiments in 200 mobile scenarios show that, our platform's FST switching latency from 60 GHz to WiFi takes 1.2 to 11.7 seconds (*c.f.* Figure 3(d)). The median latency is 3.5 seconds, which is prohibitive for delay-sensitive applications. While an aggressive timeout may improve the switching latency, it will still suffer from high oscillation between interfaces. Besides, in a reactive approach, such timeout needs to be at least greater than the total time for 60 GHz link convergence.

Summary: High switching latencies ($\geq$ 10 ms) and reactive FST-trigger approaches can severely limit the adoption of 60 GHz in today's enterprise settings. *A proactive switching algorithm that can achieve sub-10 ms latency in switching is required for seamless, high-throughput connection.*

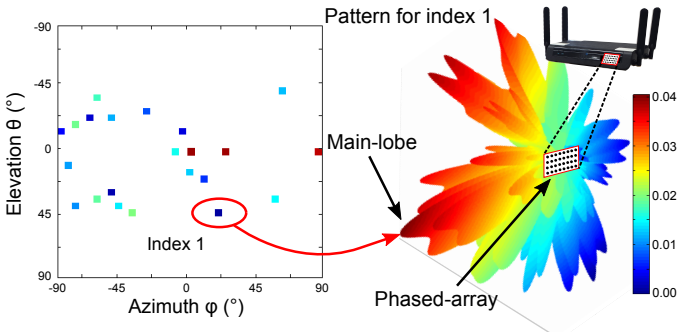

Figure 5: (a) Phased-array beam directions. (b) Measured 3D radiation pattern of beam index 1 ($G_1(\phi, \theta)$).

4 MUST DESIGN

MUST is an 802.11-compliant system that provides seamless, high-speed connectivity in dynamic indoor wireless networks. To overcome the limitations of existing designs, *MUST* sets 3 design goals. (1) *Fast 60 GHz link adaptation: MUST* seeks to instantaneously identify the best link setting at 60 GHz (beam and PHY rate) with zero probing overhead, thus evading the high link convergence time. (2) *Seamless, low-latency interface switching: MUST* aims to *proactively* migrate flows to WiFi, when 60 GHz link establishment is not feasible due to blockage, ensuring robust connectivity. It targets for seamless, sub-10 ms interface switching latency, to prevent disruptions of latency-sensitive applications. (3) *Standard compatibility:* Commodity 60 GHz and WiFi devices provide limited access/control of low-layers, and the protocol/hardware design represents a compromise between cost and performance. *MUST* thus needs to be lightweight and respect the standard constraints.

In the next sections, we describe *MUST* link adaptation (Section 4.2.1) and seamless interface switching algorithms (Section 4.2.2). We first start with a primer on 60 GHz phased-array beamforming on off-the-shelf platforms.

4.1 A Primer on Phased-Array Beamforming

A phased-array consists of a set of quasi-omni planar antenna elements arranged in a geometrical pattern (linear or rectangular). Different sets of antenna configuration weights are applied to the elements to form directional beam patterns. For example, consider a 1-D linear array of N antenna elements (with uniform separation

of d between them) that can generate K beam patterns. The pattern of the k^{th} beam is characterized by its *array-factor* (gain at azimuth direction ϕ) [56] and can be expressed as:

$$G_k(\phi) = \sum_{n=1}^{N} \mathbf{c}(n,k) \cdot e^{(j2\pi n d \cos\phi/\lambda)} \quad (1)$$

where λ is the wavelength of the wireless signal and $\mathbf{c}(n,k)$ denotes the configuration weight (discrete phase and amplitude) applied to the n^{th} element to generate k^{th} beam. The weights are designed to boost the signal strength towards a desired azimuth direction through phase-construction, while canceling signals towards unintended directions [56]. Note that, each of the *beam pattern* $G_k(\phi)$ is known and fixed once the phased-array antenna and weights are designed.

Since our 60 GHz AP platform supports a 2-D phased array (4×8 matrix), it allows directional beam generation towards various azimuth (ϕ) and elevation (θ) directions, each with gain pattern $G_k(\phi, \theta)$. For example, Figure 5(a) shows the strongest directions (*i.e.* main lobe) for a few beams in our AP. Each dot in the figure represents the desired direction of the beams (obtained from vendor's datasheet), where azimuth 0° and elevation 0° represent the orthogonal direction to the phased-array antenna's plane. Different from common perception, the beams generated via discrete antenna configurations cannot achieve perfect cone shapes. To keep the 60 GHz hardware design simple and cost-effective [47, 56, 61], practical phased arrays only employ limited phase granularity control on each element[5], which limits their ability to appropriately cancel signals towards unintended directions. To understand this issue, we followed the procedure in [34] and used an off-the-shelf 60 GHz signal strength sniffer (Vubiq PEM-003 with 3° horn antenna [36]) to measure the beam radiation pattern. We connect the AP to a 3 m away user in open LOS and place the sniffer 1.5 m from the AP. We further modified the 60 GHz kernel driver in the AP to use only transmit beam index 1 for communication. Then, we rotate the AP along azimuth and elevation plane at 3° interval. Along each interval, we generate saturated TCP traffic and measured strength of emitted signal from AP. Figure 5(b) shows measured 3D beam pattern for the beam index 1. Although the main lobe points to desired 20° azimuth and 45° elevation, there are multiple side-lobes that emit strong spurious signals to several unintended directions.

Phased-array and 60 GHz channel interaction. 60 GHz wireless channel is known to be sparse, consisting of a single dominating LOS path, and very few weak reflection paths between the transmitter and receiver [9, 44, 46, 51, 54, 60]. The former is often orders of magnitude stronger [47, 54, 64].

The phased-array beam pattern (*e.g.* Figure 5(b)) simply acts as a directional amplifier of the 60 GHz channel. When a peak lobe of the beam $G_k(\phi, \theta)$ aligns with the dominating path, the strongest channel can be created. So, the best beam k^* is the one that amplifies the channel either through its main lobe or side lobes, *i.e.*,

$$k^* = \underset{k \in \{1 \ldots K\}}{argmax} \quad \sum_{\phi,\theta} G_k(\phi,\theta) \cdot D(\phi,\theta) \quad (2)$$

[5]Phased-array on our AP allows 2-bit phase control per elements that can generate 4 possible phase values *i.e.* $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$.

where $D(\phi, \theta)$ denotes the complex channel gain along azimuth ϕ and elevation θ, *w.r.t.* AP's phased-array. Whenever the dominating path between AP and user changes either due to mobility or blockage, they need to find the new best beam k^*.

4.2 Predictable 60 GHz Performance

4.2.1 Tracking the Best Beam

Unlike existing millimeter-wave channel estimation approaches [6, 47, 54], the key goal of *MUST* is to track the dominating path between the 60 GHz transmitter and receiver, without requiring any additional 60 GHz probing, or any modifications to the off-the-shelf hardware. *MUST* predictability stems from the observation that other paths, when present, are order of magnitude weaker than the dominating path [47, 54, 64]. Thus, by simply tracking the change in the dominating path, *MUST* can predict the best beam direction and strength with reasonable accuracy. *MUST* further employs an error tracking mechanism for the prediction and LOS path blockage detection in order to maintain robust connection.

To understand how the prediction works, suppose the peak dominating path lies at azimuth ϕ and elevation θ. The channel gain at (ϕ, θ) is $D(\phi, \theta)$. Then, consider that the device's motion causes the dominating path to shift by $(\Delta\phi, \Delta\theta)$ *w.r.t.* AP. As long as we can reliably estimate the shifts, we can predict the best beam by convolving the known beam patterns $G_k(\phi, \theta)$ with the new shifted dominating path. Formally, denote $D'(\phi + \Delta\phi, \theta + \Delta\theta)$ as the channel gain of the new dominating path, then *MUST* can find the best beam as:

$$k^* = \underset{k \in \{1 \ldots K\}}{argmax} \quad \sum_{\phi,\theta} G_k(\phi,\theta) \cdot D'(\phi + \Delta\phi, \theta + \Delta\theta) \quad (3)$$

But *how to identify the shifted 60 GHz channel* $D'(\phi + \Delta\phi, \theta + \Delta\theta)$*, given that we can only measure the channel response of the current beam in use?* One possible solution is to use the out-of-band channel information from the WiFi antennas co-located in the same AP, as proposed in BBS [35]. At a high level, BBS uses WiFi CSI to estimate the absolute (ϕ, θ), which corresponds to the angle with the dominating path. Then, it selects the best beam whose main lobe is geometrically closest to (ϕ, θ). But BBS assumes a one-to-one mapping between the angular channel profile of the WiFi and 60 GHz channels, which works fine when 60 GHz devices use mechanical horn antennas [35]. However, this assumption does not hold in practical 60 GHz systems that use phased-array antennas. This is because, the 60 GHz beam direction closest to the dominating path does not necessarily yield the strongest channel. For example, the signal strength map in Figure 6(a) (top) computed by MUSIC [49] from WiFi CSI samples collected in our 3-antenna AP shows the dominating path at azimuth -64° and elevation 2°. However, the best 60 GHz beam lies at azimuth -87° and elevation -11°, as shown in Figure 6(c) (bottom), where each dot represents the main direction of the 60 GHz beam and color is its signal strength. This is because the dominating path of 60 GHz channel may not completely align with any of the available beams' main lobes, but may align with arbitrary side-lobes (Figure 5(b)), which creates the best beam spatially away from dominating direction. Also, searching for the best beam within a cluster of angular positions specified by WiFi's angular profile may not yield the best beam either.

MUST adopts a different principle: it tries to identify the angular shift of the 60 GHz dominating path from the successive time-domain spatial snapshots of the WiFi channel, and then find the 60 GHz best beam using Eq. (3). Following the foregoing example, Figure 6(b) shows the WiFi signal strength map when the device's azimuth shifts by 30° *w.r.t.* AP. Then, the successive analysis of the WiFi spatial snapshots (Figures 6(a) and 6(b)) can be used to potentially estimate the shift of 30° in the azimuth direction. Specifically, if W_1 denotes the WiFi angular profile at time t_1 and W_2 at time t_2, then the device's angular shift (equivalent to the shift of dominating path) is:

$$\{\Delta\phi, \Delta\theta\} = \underset{\Delta\phi,\Delta\theta}{argmin} \quad |W_1(\phi,\theta) - W_2(\phi+\Delta\phi, \theta+\Delta\theta)|^2$$

i.e., the angular shift can be derived from the translation of the second angular profile W_2 that best matches the first one W_1. In order to avoid abrupt error from measurement of WiFi spatial channel, *MUST* applies a time-domain linear filter on the estimated azimuth shift (with α = 0.3), *i.e.*,

$$\Delta\phi(t+1) = \alpha \cdot \Delta\phi(t) + (1-\alpha) \cdot \Delta\phi(t+1)$$

A same filter is applied on the estimated elevation shift as well. Note that, the shift of the dominating direction is not enough to find the best beam, since the strength of the dominating direction may also have changed. *MUST* applies a "reverse engineering" model on the current beam's measured gain, to detect such gain change. Specifically, if $|h_k|$ is the measured gain of the current beam k, and $|h_k^m|$ is the modeled response with the shifted dominating path, then *MUST* finds the new gain of the dominating path:

$$\begin{aligned} D'^*(\phi+\Delta\phi, \theta+\Delta\theta) &= \underset{D'}{argmin} ||h_k| - |h_k^m||^2 \\ |h_k^m| &= |\textstyle\sum_{\phi,\theta} G_k(\phi,\theta) \cdot D'(\phi+\Delta\phi, \theta+\Delta\theta)| \end{aligned} \tag{4}$$

Said differently, with the shifted dominating direction, *MUST* searches through the possible gain change in the direction by modeling the current beam's gain and matching with the measured gain. Finally, after estimating $D'^*(\phi+\Delta\phi, \theta+\Delta\theta)$ it applies Eq. (3) to identify the best beam and the SNR.

Dealing with multiple dominating paths. Our design builds upon the fundamental principle of 60 GHz channel sparsity [9, 46, 51, 54, 57] and assumes one dominating LOS path, while the side paths have order of magnitude lower signal strength. The same assumption has been adopted and validated by existing beam searching algorithm [47]. Occasionally, in open LOS, there could be additional 1 or 2 paths from strong reflectors near to the device (*e.g.* side concrete wall). Since, *MUST* uses an approximate single path model to find the best beam, it may result in inaccurate prediction. Thus, *MUST* introduces an error tracking mechanism in its prediction. Specifically, it tracks the difference between the current beam's gain h_k with the predicted gain h_k^m as: $\epsilon = ||h_k| - |h_k^m||^2$ (from Eq. (4)). When our prediction error exceeds a threshold ($\epsilon > E_t$) – *e.g.* in the presence of multiple dominant paths – then *MUST* will trigger an IEEE 802.11ad scan to recover the best beam. We empirically set the threshold E_t to 1.5 dB, which is the average SNR separation between two PHY rate options [24]. Our model has limitation to find the best beam, when

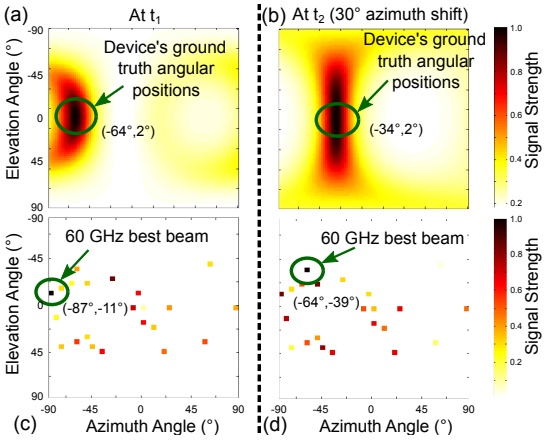

Figure 6: WiFi (top) and 60 GHz (bottom) spatial maps.

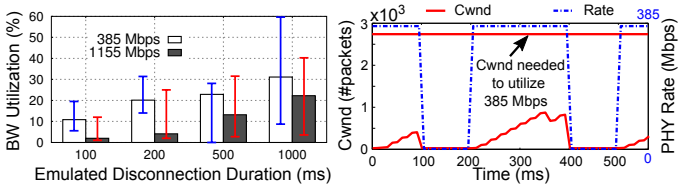

Figure 7: TCP's reaction under occassional disconnections: (a) **BW utilization.** (b) **Congestion window evolution.**

60 GHz LOS path is blocked. Typically, in such cases, 60 GHz performance drops by multiple Gbps, with rampant link disconnections. *MUST* devises a proactive blockage adaptation scheme to address this challenge, as we discuss next.

4.2.2 Proactive LOS Blockage Adaptation

When LOS blockage occurs at 60 GHz, multiple reflection paths may exist, but even the beam with the strongest channel *does not guarantee that the 60 GHz link can sustain the connection* (*c.f.* Section 3, [54, 64]). Thus, *MUST* isolates the NLOS case and handles it separately. It applies a probabilistic blockage model to identify a hint of LOS blockage, and switches to WiFi without wasting time, scanning for alternative 60 GHz beams.

LOS blockage reaction. The above design choice is based on our measurement observation that, during LOS blockage, even if 60 GHz link may intermittently offer higher PHY rate than WiFi, it is highly unstable and can have rampant disconnections with zero throughput lasting several hundred ms. Such instability is more harmful to higher layer protocols than a mediocre or even congested but stable WiFi link.

To understand this issue in detail, we set up a controlled experiment, where the AP sends saturated TCP traffic to a user over 60 GHz. Then, we emulate occasional 60 GHz link disconnections by dropping packets for a controlled time-duration at the radio. Figure 7(a) shows the effective bandwidth utilization (*i.e.* ratio of the achieved throughput over the max throughput) of TCP, under such disconnections. For occasional 100 ms disconnections, TCP's bandwidth utilization is hardly 10%, when 60 GHz rate varies from 385 Mbps (minimum PHY rate at 60 GHz) to 0 Mbps. The utilization further drops below 2%, when the rate shows a large drop from 1155 Mbps to 0 Mbps.

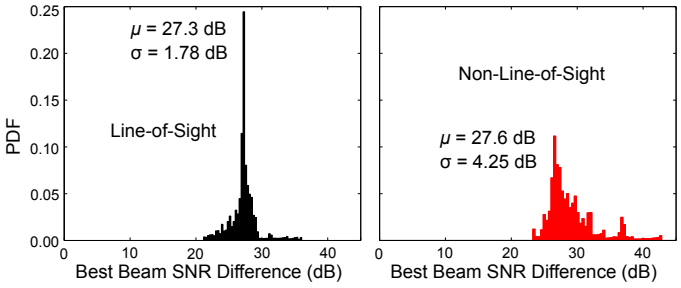

Figure 8: PDF of SNR difference between 60 GHz best beam and WiFi best path: (a) 300 LOS; (b) 150 NLOS locations.

Such poor performance in NLOS is attributed to TCP's congestion window adaptation (Figure 7(b)), which does not get enough time to explore the available bandwidth when the link becomes open. For example, Figure 7(b) shows that to utilize the 385 Mbps open link, the congestion window need to converge to approximately 2900 packets. However, link gets fully disconnected by the time it reaches hardly 800 packets. We next show how to proactively detect such blockage and switch to WiFi, to achieve a smooth real-time response at the upper layer.

LOS blockage detection. *MUST* LOS blockage detection scheme is based on the findings in [10, 53], which show that the same obstacles and NLOS reflectors have a distinct effect on signal strength change of 60 GHz and WiFi because of the frequency difference. This is also resonated in our measurements, presented in Figure 8. When the LOS is open, the signal strength difference distribution (shown in Figure 8(a)) between 60 GHz best beam and WiFi best path closely matches the hardware power budget difference of 27 dB that includes transmit power, beamforming gain and noise power difference between 60 GHz and WiFi interface. However, when the LOS is blocked, the difference shows higher variations (Figure 8(b)). This is because: (1) The same obstacle attenuates the signal strength of 60 GHz and WiFi differently, and (2) The same NLOS reflector causes distinct reflection loss on 60 GHz and WiFi signal, due to the disparate wavelengths and penetration/scattering properties [10].

MUST can identify LOS blockage by comparing the SNR between the 60 GHz beam and the best WiFi path. Once strong blockage occurs, the SNR difference diverges significantly from the hardware link budget difference, creating a signature for such blockage effect. Algorithm 1 formalizes the detection procedure. Specifically, *MUST* tracks the difference of the SNR of 60 GHz best beam and WiFi best path for a small interval Δt. To calculate the best path strength at WiFi, *MUST* selects the CSI of the WiFi antenna that is physically closest to the 60 GHz phased-array antenna, and uses the strongest tap strength of its power-delay profile. Ideally, in open LOS this difference should follow the fixed hardware link budget difference with a high probability. *MUST* thus tracks this difference and declares *LOS blockage* when the difference rises beyond a threshold σ (empirically set to 3 dB). To prevent false blockage detection, *MUST* further monitors the 60 GHz PHY rate, where an average rate below minimum PHY rate is an additional blockage hint.

Addressing false switching. *MUST* blockage detection scheme may falsely trigger interface switching. For example, Algorithm 1 may

Algorithm 1 *LOS Blockage Detection*

1: Parameters: α = 385 Mbps; Δt = 5 ms; μ = 27 dB; σ = 3 dB;
2: [] —— *During each interval* Δt *I do:*
3: Best beam SNR $B_k \leftarrow$ *MUST* prediction;
4: Best path SNR P_k of WiFi $\leftarrow$ *max IFFT*(CSI_{WiFi});
5: C $\leftarrow$ C $\cup$ $|\mu - |B_k - P_k||$;
6: Measure average PHY rate $\widehat{R}$;
7: [] —— *At the end of interval I do:*
8: **if** $\widehat{R} < \alpha$ **and** *mean*(C) $> \sigma$ **then** Switch to WiFi;
9: **else** C $\leftarrow \Phi$; Continue from 2;

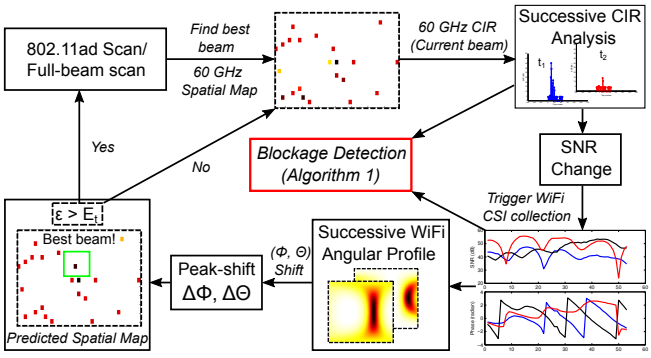

Figure 9: *MUST* prediction flow.

falsely switch a user to WiFi, when the 60 GHz dominant LOS path is blocked, while reflection paths from strong reflectors allows 60 GHz to outperform WiFi. Further, such false positive may also occur when multipath at WiFi cancel each other and the difference between 60 GHz beam and WiFi best path becomes significantly lower than the hardware link budget. However, *MUST* can rapidly recover from false positives, by intermittently performing IEEE 802.11ad beam scan while serving the user at WiFi. When IEEE 802.11ad scans indicate high 60 GHz performance, *MUST* will switch user to 60 GHz. Furthermore, we will later show that average performance loss due to the false switching is very low (less than 6% on average) mainly due to the fact 60 GHz is worse than WiFi most of the time, when the LOS is blocked.

4.3 Practical Interface Switching

Leveraging its prediction model, *MUST* allows the 60 GHz link to *instantaneously* converge to the best beam and rate setting and *proactively* switch to the WiFi interface before the link suffers from catastrophic degradation. Figure 9 summarizes the entire prediction flow for *MUST*. At run-time, *MUST* starts with existing IEEE 802.11ad beam scan to identify the single dominating LOS path and use the best beam for communication with the user device. When the performance of the current beam in use changes (*e.g.* as indicated by SNR change of the beam), *MUST* triggers to collect the out-of-band WiFi channel information. It sends probe packets to the user from the WiFi interface and measures the channel response of the received ACK. Since a probe packet is small (less than 50 bytes), the average time to probe and collect WiFi CSI from ACK is typically 1∼2 ms. Then, *MUST* uses its prediction model to identify the best beam direction and strength. Further, it employs the error tracking mechanism to

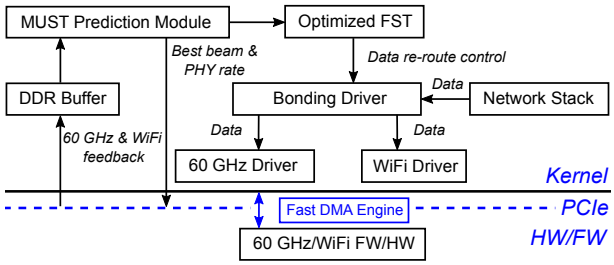

Figure 10: *MUST* system architecture.

recover from inaccurate prediction. In parallel, *MUST* employs Algorithm 1 to detect LOS blockage at 60 GHz and switch to WiFi. While operating at WiFi, *MUST* relies on existing IEEE 802.11ad beam-searching to switch traffic flows to the 60 GHz interface. Current *MUST* design is focused on AP-side adaptation, since the enterprise networks are mainly downlink (AP to user) dominated. Further, enterprise 60 GHz user devices use quasi-omni broad beams for data reception to reduce sensitivity. However, a similar adaptation can also be applied from the user side to adapt its own beam direction.

Optimizing FST. The actual flow migration between the 60 GHz and WiFi interface is realized by adopting and optimizing the IEEE 802.11ad FST feature (*c.f.* Section 2.2), which can avoid breaking the existing connections. Our platform implements IEEE 802.11ad FST over Linux's bonding driver [16], to expose a single virtual IP/MAC address towards the network stack. While the entire FST protocol to switch from 60 GHz to WiFi (including coordination and handshake) can finish within 1.5-2 ms, our measurements show that the platform requires approximately 180 ms (*c.f.* Section 3.4.2) for such operation. This is for two reasons: (1) On disconnection from 60 GHz RF front-end, current under-optimized firmware-kernel implementation still takes additional median 120 ms for message exchanges, hardware and software queuing delays and trigger FST daemon; (2) On trigger and completion of FST protocol, the daemon takes additional 60 ms to complete the virtual MAC switch at the Linux bonding driver, and reroute the packets to WiFi interface. This latency further degrades TCP's end-to-end latency (Section 3.4.2). We optimize the FST daemon and bonding driver to overcome this latency. Specifically, we generate software interrupt to the FST daemon, when Algorithm 1 detects the signature of blockage without waiting for RF disconnection from 60 GHz firmware. We further modified the bonding driver and use WiFi as backup slave instead of rehauling the virtual MAC of the active slave (*i.e.* 60 GHz). We will evaluate the effectiveness of our optimized FST with *MUST* proactive switch in Section 6.1.3. Note that, both the 60 GHz and WiFi radios employ their own power adaptation and sleep scheduling to reduce total power consumptions. An aggressive power adaptation from the user device may affect the completion time of IEEE 802.11ad FST control and coordination handshake procedure, though we found little variation throughout our experiments. We leave the detailed study of the power adaptation on *MUST* system design as our future work.

5 IMPLEMENTATION

We implement *MUST* on the commodity 60 GHz and WiFi AP without any modification to the end-user devices. Our implementation spreads across our AP's kernel and firmware, as shown in Figure 10. *MUST* software architecture is built on top of the existing open source OpenWrt [4] design, while the hardware architecture of our platform including the 60 GHz RF front-end (Section 3.1) is similar to all existing off-the-shelf tri-band platforms (*e.g.* [30]). Thus, our *MUST* design and implementation can be directly extended to any off-the-shelf tri-band platforms. The most important modifications are as follows: (1) We added suitable hooks to both kernel and firmware sides to read periodic stats from both 60 GHz and WiFi radios, and extract them in DDR buffer. The stats feed our *MUST* prediction module. We further launch adaptation commands to commit the selected beam and rate, to the 60 GHz firmware and hardware. (2) We modified our AP's existing FST daemon to handle the software interrupt from Algorithm 1, and appropriately modified the Linux's bonding driver to incorporate the practical interface switching. Note that, our platform does not allow us to trigger IEEE 802.11ad scans efficiently in real-time, which is required by our prediction algorithm. Hence, we compute the best beam and rate offline and then use the selected setting for our evaluation. We further replayed the PHY layer traces and prediction results in our platform to evaluate the switching algorithm, optimized FST and bonding driver implementation in real-time.

CPU core affinity: *MUST* runs on a multi-radio, multi-core CPU system architecture. Each wireless interface triggers interrupt request (IRQ) to notify the CPU, when a packet need to be served. The packets are copied in the kernel and user buffer for further processing. IRQ and packet processing affinity to CPU cores are critical to achieve optimal system's performance. *MUST* implements CPU core affinity based on the principle of minimizing communication required among cores, while keeping the cache hit ratio high for IRQ and packet processing modules' code and data [28]. Thus, *MUST* serves 60 GHz and WiFi packets at different cores, while assigning both IRQ and packet processing modules of an interface in the same core. We illustrate the importance of such "balanced" core affinity, by comparing it with an allocation where both 60 GHz and WiFi packet queues are processed by the same core and all IRQs are handled by another core. In our experiment, we generate saturated TCP traffic from an AP to a user at 60 GHz, while WiFi remains idle. Figure 11(a) shows the throughput (averaged over 500 ms intervals) distribution for "balanced" and "unbalanced" core affinity, where median throughput improvement is 47.3% under the former. Further, "balanced" distribution allows for more efficient CPU utilization and IRQs servicing. Figure 11(b) shows the CPU utilization and the IRQs serviced at 60 GHz, where "balanced" distribution improves utilization by average 12.8% and IRQ servicing by 54.8%.

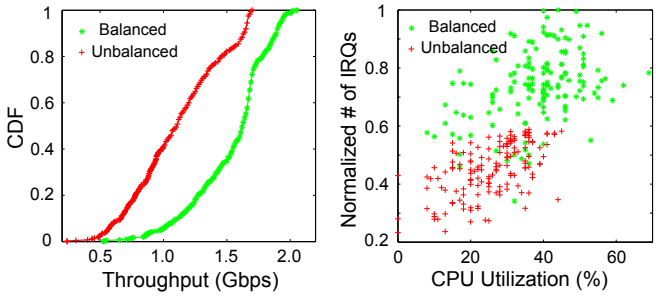

Figure 11: Effect of optimizing core affinity. (a) **Throughput performance.** (b) **CPU utilization and serviced IRQs.**

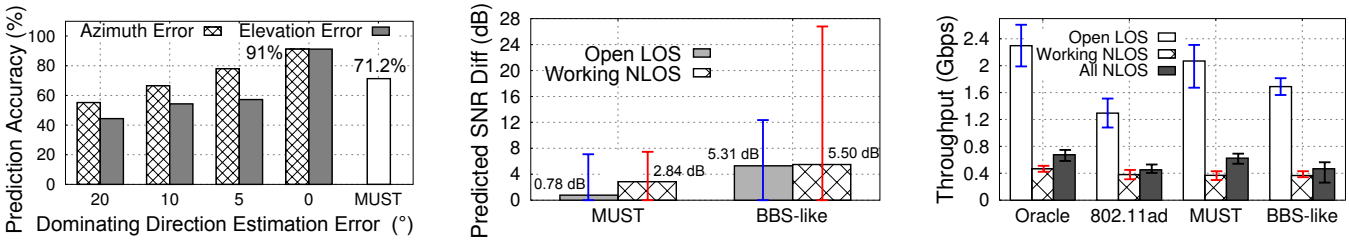

Figure 12: (a) **Best throughput beam prediction under dominating path estimation error and with *MUST*.** (b) **Predicted SNR difference of *MUST* and BBS under open LOS and working NLOS *w.r.t.* Oracle.** (c) **Throughput comparison under LOS (static and mobile), working NLOS (*i.e.* 60 GHz works) and all NLOS.**

6 EVALUATION

In this section, we evaluate *MUST* in various settings using testbed experiments and large-scale trace-driven emulation. We compare *MUST* with our platform's beam and rate selection algorithm (named as "802.11ad"), and with BBS [35]. We also compare *MUST* prediction with an "Oracle" solution which instantaneously converges to the best setting.

6.1 Micro-benchmarks

We first evaluate *MUST* along *three* key performance metrics: (1) How *accurately MUST* can predict the best quality beam? (2) How much *performance improvement MUST* can bring to 60 GHz link? (3) How *accurately MUST* can detect blockage and how *fast* it can switch from 60 GHz to WiFi?

6.1.1 Prediction Accuracy

Identifying best beam. We first evaluate how accurately *MUST* can zero in on the best quality beam, defined as the one which maximizes receiver's SNR. For our experiments, we mounted our AP on a 9 ft high tripod in the same enterprise setting, and placed a user device within 10×10 m^2 area (Figure 1). Then, we compare *MUST*'s beam prediction with Oracle, in multiple static, LOS and NLOS scenarios. We find the Oracle (optimal) beam, by scanning every transmit beam for each static point, and pick the one with highest receiver's SNR. Figure $12(a)$ shows that *MUST* can predict the best beam with 71.2% accuracy compared to Oracle. The root causes for inaccurate prediction are twofold. First, *MUST* may not always precisely track the 60 GHz dominant path, and may end up converging to suboptimal beams. We investigate how such tracking errors affect prediction accuracy, by artificially introducing errors (*i.e.* shift in azimuth and elevation direction) in the measured dominating direction which varies from 5° to 20°. Figure $12(a)$ shows even an error of 10° can drop the overall prediction accuracy by over 30%. This is because the dominating direction itself typically spans over a small angular spread of $<10^\circ$ [60]. Second, *MUST* approximates the entire 60 GHz channel using a single dominating direction. Even an accurate dominant direction tracking (0° error) that models single path, can at best achieve 91% accuracy. However, *MUST* error tracking and blockage detection mechanisms can recover such inaccurate predictions.

Beam quality prediction. We further evaluate how accurately *MUST* can predict user's SNR compared to Oracle, for the LOS and NLOS settings described above. Figure $12(b)$ shows that in LOS, *MUST* average prediction error lies within 0.78 dB. Since, 60 GHz PHY rate options are on average separated by 1.5 dB signal strength, this error has a negligible effect on achievable throughput performance of *MUST*, as we will show later. We further compare BBS [35] under the same setup. Since, BBS can not directly predict the signal strength, we calibrate the WiFi peak direction signal strength with the hardware link budget difference between 60 GHz and WiFi. BBS's prediction error is significantly higher and exceeds 5 dB on average for both LOS and NLOS settings, resulting in erroneous rate selection. In NLOS, the error from *MUST* prediction can be higher (average 2.84 dB), however, in most NLOS cases it will trigger IEEE 802.11ad scans to recover from such errors.

6.1.2 Throughput Gain from Prediction

We next evaluate the throughput gains of *MUST* over 802.11ad and BBS achieved from selecting best beam and rate. Since BBS does not specify any rate adaptation algorithm, we use our platform's default one. To perform a fair comparison, we assume that BBS will switch to WiFi when 60 GHz PHY rate drops below minimum PHY rate. Figure $12(c)$ shows that *MUST* achieves on average 25% and 60% gains over BBS and 802.11ad in LOS, while it remains close to 10% to Oracle. In working NLOS cases (where 60 GHz is still the best interface), *MUST* performs similar to BBS and 802.11ad, and achieves within 20% from the Oracle. This is because in such scenarios, the existence of multiple dominating paths may lead to inaccurate beam and rate prediction, and thus *MUST* will frequently fallback to 802.11ad. In all NLOS, including where 60 GHz links are blocked, all algorithms will eventually switch to WiFi. *MUST* can almost instantaneously switch to WiFi achieving 34% and 39% throughput gains over BBS and 802.11ad, respectively. Note that, original BBS as described in [35] does not have such interface switching capability, and thus in practice it will achieve zero throughput, when 60 GHz links break.

6.1.3 Blockage Detection and Fast Switching

We now evaluate efficacy of *MUST* to detect LOS blockage (Algorithm 1) and switch to WiFi.

Accuracy and false detection. We conduct a simple experiment with TCP traffic from an AP to a user device. The user moves along a fixed trajectory for 15 seconds within the 10×10 m^2 area. The trajectory consists of approximately 1/3 blockage and 2/3 open LOS to the AP. Since, finding the optimal performing interface in mobile scenario is difficult, we emulate the mobility by statically positioning the user device along the fixed trajectory and measure the maximum throughput achieved from both 60 GHz and WiFi interfaces. Then we run Algorithm 1 to decide the best performing interface. We define accurate decision as the cases in which *MUST* chose the highest

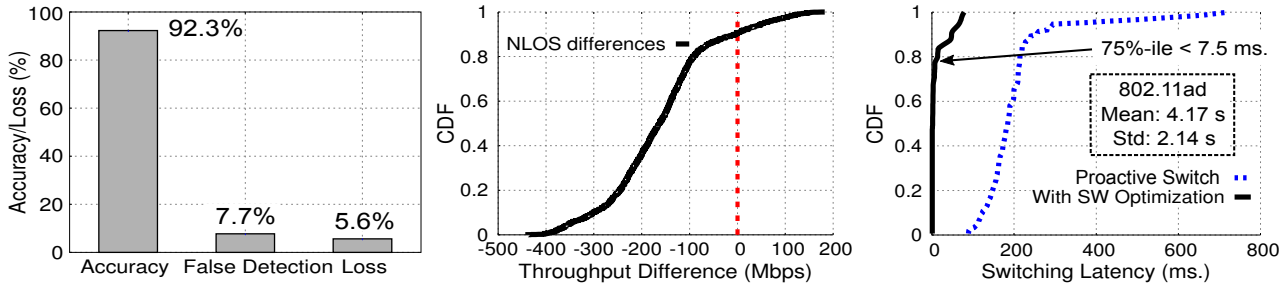

Figure 13: Blockage detection & switching: (a) Accuracy, false detection and throughput loss. (b) Throughput difference between 60 GHz and WiFi in all NLOS. (c) Interface switching latency under proactive switch and with software optimization.

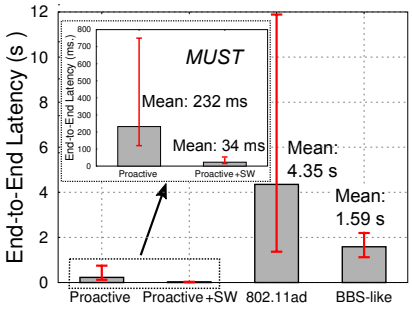

Figure 15: TCP end-to-end latency with *MUST*.

throughput interface. Figure 13(a) shows that in 92.3% of the cases, *MUST* can accurately detect the best performing interface. Although, the heuristic nature of Algorithm 1 can result in average 7.7% false detections, the average throughput loss is below 6% compared to Oracle. This is because, in NLOS, 60 GHz performs worse than WiFi most of the times. Specifically, we observe that in less than 10% of the NLOS (out of more than 200 NLOS locations), 60 GHz outperforms WiFi as shown in Figure 13(b). *MUST* never falsely switches to WiFi in open LOS, since it monitors the 60 GHz PHY rate, which remains well above 385 Mbps in LOS (*c.f.* Algorithm 1).

Switching latency improvement. Blockage detection triggers optimized FST of *MUST*. We next evaluate the switching latency of our fast switching (described in Section 4.3) in multiple blockage scenarios. Figure 13(c) shows that switching latency is bounded by 7.5 ms in 75%-ile of the cases with standard deviation of 1.46 ms. The latency deviation is attributed to our platform's OS and firmware related idiosyncrasies[6]. We believe this latency can be further minimized with tighter control over 60 GHz firmware which is currently unavailable in the platform. *MUST* achieves almost 2 orders of magnitude improvement from off-the-shelf 802.11ad solution that has an average 4.17 s switching latency.

TCP performance under fast switch. The proactive, low-latency FST of *MUST* design brings benefit to the higher layer applications. We measure TCP's end-to-end latency for *MUST*, 802.11ad and BBS. End-to-end latency is measured as *RTT*/2 for the single hop connection. Figure 15 shows that *MUST* proactive interface switching (without the FST software optimization) reduces the 802.11ad solution's average TCP end-to-end latency from 4.35 s to 232 ms. The

[6]Proactive switch without the software optimization still suffers from high latency (median 180 ms) due to the off-the-shelf platforms' poor driver implementation (*c.f.* dotted line in Figure 13(c)).

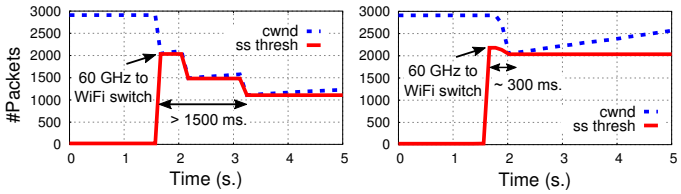

Figure 16: TCP congestion window and slow start threshold under: (a) Proactive switch. (b) With software optimization.

software optimization brings additional benefit to *MUST*, and further drops the average end-to-end latency to 34 ms. The additional benefit is further reflected in TCP's congestion window adaptation. Specifically, Figure 16(a) shows that, the additional latency introduced by the non-optimized switch, will increase TCP dropped packets, and will affect the TCP congestion window and slow start threshold. While under an optimized switch (Figure 16(b)), congestion window can quickly start to converge within 300 ms, non-optimized implementation introduces larger delay ($>$ 1.5 s).

6.2 Field Trials

Our experiments so far have been focused on single AP-user settings. We next evaluate *MUST* in more realistic field trials conducted in an enterprise (Figure 1) and campus buildings, where various sources of dynamics including environment mobility and WiFi interferences coexist in a complex manner.

TCP throughput. Figure 14(a) compares *MUST*, BBS and 802.11ad in settings where an AP generates saturated downlink TCP traffic to a varying number (one to three) of static and mobile users with pedestrian walking speed (xU/yM implies y mobile out of the total x users). Our results show that *MUST* throughput gains increase during users' mobility. When all the users are mobile (1U/1M, 2U/2M & 3U/3M), *MUST* achieves on average 45% and 48% throughput gains over BBS and 802.11ad, respectively. However, for static LOS users, BBS and 802.11ad can converge to the best beam and rate setting, reducing *MUST* gains.

We further elaborate on *MUST* gains under mobility, by monitoring a user's TCP throughput in a 15 s mobile trace, presented in Figure 14(b). Initially, all the algorithms operate on 60 GHz and achieve more than 1.6 Gbps average throughput. Between 6^{th} and 6.5^{th} second, 60 GHz link breaks. While *MUST* instantaneously switches to WiFi, BBS and 802.11ad require roughly 3.5 s to switch interface, achieving zero throughput during the switch.

UDP gains. We further compare *MUST* with BBS and 802.11ad in our field-trial settings with saturated UDP traffic. In order to effectively

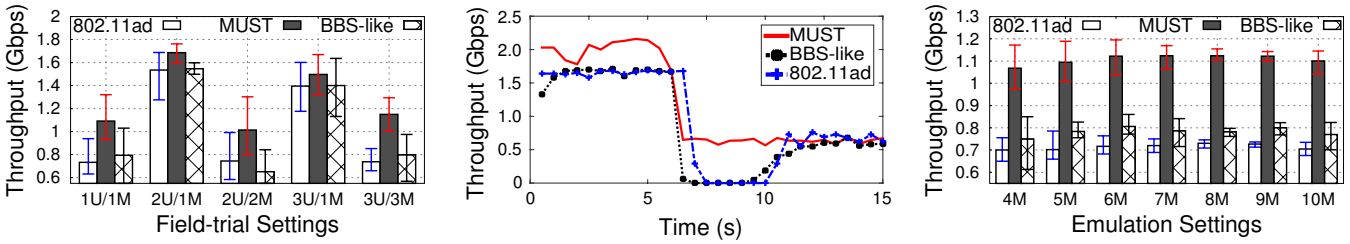

Figure 14: *MUST*, BBS and 802.11ad comparison in field-trial and trace-driven emulation: (a) **Field-trial throughput with static and mobile users.** (b) **Example mobility trace for a 15 s walk (pedestrian speed).** (c) **Emulated multi-user throughput performance.**

Algorithm Comparison	Gain (%)		
	Median	Max	Min
vs. 802.11ad	49.3	58.0	43.8
vs. BBS-like	45.6	57.7	28.3

Table 1: *MUST* UDP gains in the field-trial settings.

use the Gbps 60 GHz link speed, we configure the UDP settings at both AP and user side to support jumbo frames and larger transport layer socket buffer following [33]. *MUST* achieves 49.3% and 45.6% median UDP throughput gain over the 802.11ad and BBS respectively (Table 1). Similar to TCP, *MUST* achieves highest gain during mobility due to faster link adaptation and WiFi switching.

Trace-driven emulation. We finally resort to trace-driven emulation to evaluate larger topologies with up to 10 users. We consider mobile users or moving obstacles in users' physical environment. Figure 14(c) shows that *MUST* achieves on average 42% and 55% TCP throughput gains over BBS and 802.11ad across all the settings. Moreover, we observe that in such dynamic environments, the number of users connected to the AP does not considerably affect *MUST* gains.

7 RELATED WORK

Millimeter-wave measurements and modeling. Properties of the millimeter-wave wireless channel, including the effects of environmental reflection and human blockage have been well studied through measurements or statistical models [9, 10, 15, 22]. It has been observed that the millimeter-wave channel is sparse, *i.e.*, there exist only a few paths between the transmitter and receiver. Such sparsity results in fragile 60 GHz links, during blockage and mobility [51, 53, 54, 64]. However, it creates a unique opportunity for a predictive model of link performance, as shown in *MUST*.

Cross-band channel prediction. Recent works [50, 55] harnessed wireless channel correlation across frequency bands to predict channel quality. For example, CSpy [50] built a machine learning model that implicitly captures such correlation to infer best quality alternative WiFi channel without probing. R2-F2 [55] enables LTE base stations to infer the downlink channel by observing the uplink channels from the same user. BBS [35] leveraged the WiFi channel to narrow down 60 GHz beam search space, which works fine for systems using mechanical horn antennas. *MUST* on the other hand can instantaneously predict the best beam and PHY rate settings for practical 60 GHz systems employing phased-array beamforming.

60 GHz link adaptation. Many millimeter-wave systems [6, 47, 54] explored ways of estimating 60 GHz beam quality, with minimum measurement overhead. BeamSpy [54] models the 60 GHz channel using a discrete set of signal paths and can instantaneously predict the best alternative beam under blockage. But it is applicable only for quasi-stationary links. AgileLink [6] hashes the beam directions, and quickly identifies the best path by tracking how the energy changes across different hash functions. However, it requires fine-grained phase control over the antenna elements, whereas standard off-the-shelf 60 GHz devices (including our platform) only support 2-4 bit phase control [47]. Compressive sensing based approaches have also been proposed for tracking the 60 GHz paths [32, 47, 62], but they require customized beaconing signals, and the high computational latency may neutralize the saving of beam searching overhead.

Multi-radio collaboration. Existing work on multi-radio collaboration at lower frequency bands has been focused on: traffic management among interfaces (*e.g.*, data offloading across LTE and WiFi) [11, 13, 29, 31, 41], mobility management [42], energy efficiency [7, 37–39], and routing in multi-band mesh networks [8, 19, 52]. Different from these efforts, *MUST* is the first one to explore the principles of coordination between two completely disparate technologies: 60 GHz and WiFi. The huge bit-rate gaps, signal coverage patterns and distinct sensitivities to blockage and mobility together render traditional solutions (*e.g.*, MPTCP, probe-and-adapt) ineffective.

8 CONCLUSION

In this paper, we explore the possibility of leveraging the existing WiFi technology to design high-performance, robust 60 GHz wireless networks. To this end, we design and implement *MUST* using standard, off-the-shelf hardware that provides seamless, high-speed connectivity in highly dynamic 60 GHz indoor environments. *MUST* introduces a WiFi-assisted 60 GHz link adaptation module that can instantaneously predict the best 60 GHz beam and PHY rate setting. It further designs a proactive switching algorithm to achieve sub-10 ms switch to WiFi, allowing latency-sensitive traffic to run seamlessly over existing network stack. Our experimental results confirm high gain of *MUST* over state-of-the-art solutions. We consider *MUST* to be a key building block of 60 GHz WLANs, and an important component of next-generation 5G networks, which will consist of dense picocellular millimeter-wave deployments, alongside WiFi/LTE.

ACKNOWLEDGEMENT

We sincerely thank the anonymous reviewers and our shepherd David Chu for their valuable comments and feedback. Sanjib Sur and Xinyu Zhang were partially supported by the NSF under Grant CNS-1617321, CNS-1506657, CNS-1518728, CNS-1343363, and CNS-1350039 during this work.

REFERENCES

[1] FCC Promotes Higher Frequency Spectrum for Future Wireless Technology. https://apps.fcc.gov. 2015.

[2] Verizon applauds FCC chairman's move to 5G spectrum. http://www.fiercewireless.com/tech/verizon-applauds-fcc-chairman-s-move-to-5g-spectrum. 2015.

[3] Unlocking the Promise of Broadband for All Americans. https://obamawhitehouse.archives.gov/blog/2016/07/15/unlocking-promise-broadband-generate-gains-all-americans. 2016.

[4] OpenWrt Wireless Freedom. https://openwrt.org/. 2017.

[5] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Cutting the Cord in Virtual Reality. In *Proc. of ACM HotNets-XV*. 2016.

[6] Omid Abari, Haitham Hassanieh, Michael Rodreguez, and Dina Katabi. Millimeter Wave Communications: From Point-to-Point Links to Agile Network Connections. In *Proc. of ACM HotNets-XV*. 2016.

[7] Yuvraj Agarwal, Trevor Pering, Roy Want, and Rajesh Gupta. SwitchR: Reducing System Power Consumption in a Multi-Client, Multi-Radio Environment. In *IEEE International Symposium on Wearable Computers*. 2008.

[8] Mansoor Alicherry, Randeep Bhatia, and Li Erran Li. Joint Channel Assignment and Routing for Throughput Optimization in Multiradio Wireless Mesh Networks. *IEEE Journal on Selected Areas in Communications* Vol. 24, No. 11. 2006.

[9] C.R. Anderson and T.S. Rappaport. In-Building Wideband Partition Loss Measurements at 2.5 and 60 GHz. *IEEE Transactions on Wireless Communications* Vol. 3, No. 3. 2004.

[10] Christopher R. Anderson, Theodore S. Rappaport, Kyung Bae, Alex Verstak, Naren Ramakrishnan, William H. Tranter, Clifford A. Shaffer, and Layne T. Watson. In-Building Wideband Multipath Characteristics at 2.5 & 60 GHz. In *Proc. of IEEE VTC*. 2002.

[11] Aruba Networks, Inc. Carrier-Class Public WiFi for 3G/4G Offload. http://www.arubanetworks.com/pdf/solutions/SB_Offload.pdf. 2012.

[12] IEEE Standards Association. IEEE Standards 802.15.3c-2009: Millimeter-wave-based Alternate Physical Layer Extension. 2009.

[13] Mehdi Bennis, Meryem Simsek, Andreas Czylwik, Walid Saad, Stefan Valentin, and Merouane Debbah. When Cellular Meets WiFi in Wireless Small Cell Networks. *IEEE Communications Magazine* Vol. 51, No. 6. 2013.

[14] Ethan Blanton and Mark Allman. On Making TCP More Robust to Packet Reordering. In *Proc. of ACM SIGCOMM*. 2002.

[15] Sylvain Collonge, Gheorghe Zaharia, and Ghais El Zein. Influence of the Human Activity on Wide-Band Characteristics of the 60 GHz Indoor Radio Channel. *IEEE Transactions on Wireless Communications* Vol. 3, No. 6. 2004.

[16] Thomas Davis. Linux Ethernet Bonding Driver HOWTO. www.kernel.org. 2011.

[17] Dell Inc. Latitude 12 5000 Series (E5250). http://www.dell.com/en-us/work/shop/productdetails/latitude-e5250-laptop. 2016.

[18] Shuo Deng, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. WiFi, LTE, or Both? Measuring Multi-Homed Wireless Internet Performance. In *Proc. of ACM IMC*. 2014.

[19] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *Proc. of ACM MobiCom*. 2004.

[20] ECMA International. Standard ECMA-387: High Rate 60 GHz PHY, MAC and PALs. 2010.

[21] Simone Ferlin, Thomas Dreibholz, and Ozg Alay. Multi-Path Transport Over Heterogeneous Wireless Networks: Does It Really Pay Off?. In *Proc. of IEEE Globecom*. 2014.

[22] Nadir Hakem, Gilles Delisle, and Yacouba Coulibaly. Radio-Wave Propagation into an Underground mine environment at 2.4 GHz, 5.8 GHz and 60 GHz. In *European Conference on Antennas and Propagation (EuCAP)*. 2014.

[23] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. Augmenting Data Center Networks with Multi-gigabit Wireless Links. In *Proc. of ACM SIGCOMM*. 2011.

[24] IEEE Standards Association. IEEE Standards 802.11ad-2012, Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band. 2012.

[25] IEEE Standards Association. IEEE Standards 802.11ac-2013: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. 2013.

[26] Injong Rhee and Lisong Xu and Sangtae Ha and Alexander Zimmermann and Lars Eggert and Richard Scheffenegger. CUBIC for Fast Long-Distance Networks. In *Internet-Draft*. 2015.

[27] Intrinsyc Software International, Inc. Mobile Development Platform Tablet. https://www.intrinsyc.com/press-release/intrinsyc-announces-development-platforms-featuring-new-high-performance-qualcomm-snapdragon-820-processor/. 2016.

[28] Hye-Churn Jang and Hyun-Wook Jin. MiAMI: Multi-core Aware Processor Affinity for TCP/IP over Multiple Network Interfaces. In *IEEE Symposium on High Performance Interconnects*. 2009.

[29] Xavier Lagrange. Very Tight Coupling between LTE and Wi-Fi for Advanced Offloading Procedures. In *IEEE Wireless Communications and Networking Conference Workshops*. 2014.

[30] TP-LINK Technologies Co. Ltd. Wi-Fi Routers | AD7200. http://www.tp-link.com/us/products/details/cat-5506_AD7200.html. 2016.

[31] Rajesh Mahindra, Hari Viswanathan, Karthik Sundaresan, Mustafa Y. Arslan, and Sampath Rangarajan. A Practical Traffic Management System for Integrated LTE-WiFi Networks. In *Proc. of ACM MobiCom*. 2014.

[32] Zhinus Marzi, Dinesh Ramasamy, and Upamanyu Madhow. Compressive Channel Estimation and Tracking for Large Arrays in mm-Wave Picocells. Vol. 10, No. 3. 2016.

[33] Energy Sciences Network. UDP Tuning. https://fasterdata.es.net/network-tuning/udp-tuning. 2012.

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

[35] Thomas Nitsche, Adriana B. Flores, Edward W. Knightly, and Joerg Widmer. Steering with Eyes Closed: mm-Wave Beam Steering without In-Band Measurement. In *Proc. of IEEE INFOCOM*. 2015.

[36] Pasternack. 60 GHz Transmit/Receive (Tx/Rx) Development System. https://www.pasternack.com. 2013.

[37] Ioannis Pefkianakis, Jaideep Chandrashekar, and Henrik Lundgren. User-Driven Idle Energy Save for 802.11x Mobile Devices. In *Proc. of IEEE MASS*. 2014.

[38] Trevor Pering, Yuvraj Agarwal, Rajesh Gupta, and Roy Want. CoolSpots: Reducing the Power Consumption of Wireless Mobile Devices with Multiple Radio Interfaces. In *Proc. of ACM MobiSys*. 2006.

[39] Christopher Pluntke, Lars Eggert, and Niko Kiukkonen. Saving Mobile Device Energy with Multipath TCP. In *Proc. of ACM MobiArch*. 2011.

[40] Qualcomm Incorporated. Techniques for Rate Selection in Millimeter-Wave Communication Systems. In *US Patent 9,312,985*. 2016.

[41] Qualcomm Technologies, Inc. LTE and Wi-Fi Convergence: Leveraging Both to Meet Capacity Demand. https://www.qualcomm.com/news/onq/2014/02/18/lte-and-wi-fi-convergence-leveraging-both-meet-capacity-demand. 2014.

[42] Costin Raiciu, Dragos Niculescu, Marcelo Bagnulo, and Mark James Handley. Opportunistic Mobility with Multipath TCP. In *Proc. of ACM MobiArch*. 2011.

[43] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. In *Proc. of USENIX NSDI*. 2012.

[44] T.S. Rappaport, F. Gutierrez, E. Ben-Dor, J.N. Murdock, Yijun Qiao, and J.I. Tamir. Broadband Millimeter-Wave Propagation Measurements and Models Using Adaptive-Beam Antennas for Outdoor Urban Cellular Communications. *IEEE Transactions on Antennas and Propagation* Vol. 61, No. 4. 2013.

[45] T.S. Rappaport, Shu Sun, R. Mayzus, Hang Zhao, Y. Azar, K. Wang, G.N. Wong, J.K. Schulz, M. Samimi, and F. Gutierrez. Millimeter Wave Mobile Communications for 5G Cellular: It Will Work! *IEEE Access* 1. 2013.

[46] Theodore S. Rappaport, Eshar Ben-Dor, James N. Murdock, and Yijun Qiao. 38 GHz and 60 GHz Angle-Dependent Propagation for Cellular and Peer-to-Peer Wireless Communications. In *Proc. of IEEE ICC*. 2012.

[47] Maryam Eslami Rasekh, Zhinus Marzi, Yanzi Zhu, Upamanyu Madhow, and Haitao Zheng. Noncoherent mmWave Path Tracking. In *Proc. of ACM HotMobile*. 2017.

[48] Grand View Research Report. Wireless Gigabit (WiGig) Market Size To Reach $7.42 Billion By 2024. http://www.grandviewresearch.com/press-release/global-wireless-gigabit-wigig-market. 2016.

[49] Ralph O. Schmidt. Multiple Emitter Location and Signal Parameter Estimation. *IEEE Transactions on Antennas and Propagation* Vol. 34, No. 3. 1986.

[50] Souvik Sen, Bozidar Radunovic, Jeongkeun Lee, and Kyu-Han Kim. CSpy: Finding the Best Quality Channel without Probing. In *Proc. of ACM MobiCom*. 2013.

[51] Peter F. M. Smulders. Statistical Characterization of 60-GHz Indoor Radio Channels. *IEEE Transactions on Antennas and Propagation* Vol. 57, No. 10. 2009.

[52] Anand Prabhu Subramanian, Himanshu Gupta, Samir R. Das, and Jing Cao. Minimum Interference Channel Assignment in Multiradio Wireless Mesh Networks. *IEEE Transactions on Mobile Computing* Vol. 7, No. 12. 2008.

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

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

[55] Deepak Vasisht, Swarun Kumar, Hariharan Rahul, and Dina Katabi. Eliminating Channel Feedback in Next-Generation Cellular Networks. In *Proc. of ACM SIGCOMM*. 2016.

[56] Junyi Wang, Zhou Lan, Chin-Sean Sum, Chang-Woo Pyo, Jing Gao, Tuncer Baykas, Azizur Rahman, Ryuhei Funada, Fumihide Kojima, Ismail Lakkis, Hiroshi Harada, and Shuzo Kato. Beamforming Codebook Design and Performance Evaluation for 60GHz Wideband WPANs. In *Proc. of IEEE VTC*. 2009.

[57] Teng Wei and Xinyu Zhang. Facilitating Robust 60 GHz Network Deployment By Sensing Ambient Reflectors. In *Proc. of USENIX NSDI*. 2017.

[58] Teng Wei and Xinyu Zhang. Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support. In *Proc. of ACM MobiCom*. 2017.

[59] www.qualcomm.com. Qualcomm Atheros and Wilocity Announce Tri-band Wi-Fi: Industry's First Standards-compliant, Multi-Gigabit Wireless

Chipset. https://www.qualcomm.com/news/releases/2011/05/31/qualcomm-atheros-and-wilocity-announce-tri-band-wi-fi-industrys-first. 2011.

[60] Hao Xu, Vikas Kukshya, and Theodore S. Rappaport. Spatial and Temporal Characteristics of 60-GHz Indoor Channels. *IEEE Journal on Selected Areas in Communications* Vol. 20, No. 3. 2002.

[61] Jialiang Zhang, Xinyu Zhang, Pushkar Kulkarni, and Parameswaran Ramanathan. OpenMili: A 60 GHz Software Radio Platform with a Reconfigurable Phased-Array Antenna. In *Proc. of ACM MobiCom.* 2016.

[62] Anfu Zhou, Xinyu Zhang, and Huadong Ma. Beam-forecast: Facilitating Mobile 60 GHz Networks via Model-driven Beam Steering. In *Proc. of IEEE INFOCOM.* 2016.

[63] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *Proc. of ACM SIGCOMM.* 2012.

[64] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y. Zhao, and Haitao Zheng. Demystifying 60GHz Outdoor Picocells. In *Proc. of ACM MobiCom.* 2014.