Practical Lower Layer 60 GHz Measurements Using Commercial Off-The-Shelf Hardware

Adrian Loch[‡], Guillermo Bielsa[‡§], and Joerg Widmer[‡]
[‡]IMDEA Networks Institute, Madrid, Spain
[§]Universidad Carlos III de Madrid, Spain
{firstname.lastname}@imdea.org

ABSTRACT

Experimenting platforms for wireless 60 GHz networking measurements are limited and extremely costly. The requirements for such a platform in terms of bandwidth and antenna capabilities are very high. For instance, the 802.11ad protocol uses channels with a bandwidth of 2.16 GHz and requires electronically steerable phased antenna arrays. Devices implementing this protocol are available as consumer-grade off-the-shelf hardware but are typically a black box which barely allows any insights for research purposes. In this paper, we show the hidden monitoring capabilities of such a consumer-grade 60 GHz device, and explain how to access lower layer parameters such as modulation and coding schemes, antenna steering, and packet decoding. Moreover, we present an extensive set of experiments showing the behavior of these parameters by means of the aforementioned monitoring capabilities.

CCS Concepts

•**Networks** → **Network experimentation; Wireless local area networks;** *Network measurement;* Physical links; •**Hardware** → *Wireless devices;*

Keywords

802.11ad; WiGig; 60 GHz networks; practical performance; lower layer measurement analysis

1. INTRODUCTION

Practical wireless network experimentation in the 60 GHz band is often infeasible. This is particularly critical at present since the first IEEE 802.11ad[1] devices are on the verge of reaching the consumer market. For instance, TP-Link is launching its Talon AD7200 multi-band router [16], and LeTV has announced the matching Le Max Pro smartphone [6] with 802.11ad support. As a result, the limitations of 60 GHz networking in practical real-world deployments are about to become evident. For the research community in this field it is fundamental to fully understand these practical limitations to be able to provide efficient solutions. From a networking perspective, such an analysis is very hard for a perplexing reason—lack of suitable hardware.

At the time of writing, companies manufacturing test and measurement equipment are providing hardware for experimentation in the millimeter wave (mmWave) band, which includes 60 GHz. For instance, National Instruments (NI) recently launched its mmWave Software Defined Radio (SDR) [8]. However, this platform is designed with a focus on the physical layer. While it allows users to implement any functionality on it, NI provides a reference design for the physical layer only. Given the complexity of (a) the Medium Access Control (MAC) layer in 802.11ad and (b) the FPGA-based architecture of NI's system, this is a major limitation. The MAC plays a critical role in 60 GHz networks. Nodes typically must use beamforming to overcome the large attenuation at such high frequencies. Hence, the MAC layer must implement beam search mechanisms to ensure proper steering towards other nodes. This has a significant impact on the performance of the network since inefficient beam search results in high overhead, and poor beam steering translates into a low Signal-to-Noise Ratio (SNR). Further, mechanisms at the MAC such as aggregation are fundamental to ensure high efficiency—if packets are not aggregated in large frames, the medium access overhead in Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) networks becomes prohibitive. A complete MAC layer implementation and its interaction with the network and the transport layers is essential to correctly assess the performance of 60 GHz networks. In other words, to understand the *network* behavior in the 60 GHz band, we must consider the full stack but also have fine-grained access to parameters of the lower layers. The latter are the key difference between mmWave networks and networks operating in lower bands.

The above requirements result in a significant challenge regarding which experimental platform to use. On the one hand, cost-efficient off-the-shelf hardware that implements the full stack is a black box when it comes to the lower layers. On the other hand, existing mmWave prototyping platforms that provide access to the lower layers lack built-in 802.11ad MAC functionalities. Moreover, such platforms are extremely costly, which leads to a limited testbed size. This massively hinders practical research in 60 GHz networks.

In this paper, we present a "golden mean" solution in between the above two extremes. Specifically, we gain access

[1]The IEEE 802.11ad standard [5] is an amendment to IEEE 802.11-2012 for wireless networking in the 60 GHz band.

WiNTECH'16, October 03-07 2016, New York City, NY, USA

ISBN 978-1-4503-4252-0/16/10…$15.00

DOI: http://dx.doi.org/10.1145/2980159.2980165

to in-depth lower layer information of a commercial off-the-shelf (COTS) 60 GHz device. Access to this information allows us to carry out experiments involving the full networking stack but at the same time also gives us detailed insights on the operation of the lower layers. While not configurable, this solution is extremely cost-efficient compared to, for instance, NI's system, allowing for a large testbed. The COTS device that we use is the well-known Dell D5000 docking station system (c.f. Section 2), which many research groups in this field have used as a basis for practical experiments. However, earlier experiments with this platform were inevitably limited to throughput measurements at the transport layer and rate measurements provided by the configuration tool of the device [18, 12]. In contrast, we enable physical and MAC layer measurements using this device. We believe that this is a very valuable tool for the 60 GHz networking research community. In particular, our contributions are as follows.

1. We introduce a methodology to gain access to in-depth information reported by the D5000 driver.

2. We present an extensive analysis of lower layer parameters in an indoor mobility scenario.

3. We show the impact of transient blockage on lower layer parameters such as beam steering.

The remainder of this paper is organized as follows. In Section 2 we survey the capabilities of existing 60 GHz hardware used in related work. After that, in Section 3 we explain the monitoring features of the D5000 device driver. Concretely, we show how to exploit them for experimentation measurements. In Section 4 we present practical measurements obtained using our approach in both mobility and transient blockage scenarios. Finally, Section 5 concludes the paper.

2. BACKGROUND AND RELATED WORK

Related work performs practical experiments in the 60 GHz band based on a number of approaches and goals. Regarding the latter, we categorize earlier work into (a) channel characterization, and (b) network evaluation. We briefly discuss (a) but then focus on (b), which is the main focus of this paper.

2.1 Channel Characterization

Work on experimental indoor channel characterization in the 60 GHz band dates back to long before the standardization of 802.11ad in this band took place. For instance, the authors of [17] perform an exhaustive analysis of the angle-of-arrival and time-of-arrival in an indoor 60 GHz environment. Based on this analysis, they derive the statistical parameters of the channel to formulate detailed channel models for a number of indoor scenarios. Recent work has extended such measurements for different mmWave bands in densely populated outdoor environments [11]. The goal is to assess the feasibility of mmWave broadband cellular communication networks. To this end, the authors use specialized measurement equipment for channel sounding at both 28 GHz and 38 GHz. Their test setup focuses on capturing the physical layer characteristics, that is, it does not transmit actual packets.

2.2 Network Evaluation

Related work uses a number of different experimenting platforms to evaluate mmWave networks in practice. We categorize them into four approaches.

2.2.1 Signal Generator and Analyzer

First, some authors use a signal generator attached to a 60 GHz up-converter as a transmitter, and a signal analyzer attached to a 60 GHz down-converter as a receiver [13, 10]. Since the converters are typically attached to horn antennas, this allows for detailed studies of the impact of directional communication. However, no data transmission takes place. Hence, such a setup allows for SNR measurements only. To provide an intuition of the data rates that would be possible at the measured SNR, researchers often use the receive sensitivity of each data rate standardized in IEEE 802.11ad to convert SNR into rate.

2.2.2 Narrowband SDRs

A second approach to 60 GHz network evaluation uses narrowband SDRs for both transmitter and receiver, instead of the aforementioned signal generator and signal analyzer [14, 4, 3]. This enables researchers to implement actual packet communication. However, the bandwidth is restricted to the capabilities of the SDR. To address this limitation, the authors of [14] extended the Wireless Open-Access Research Platform (WARP) SDR with a custom 245 MHz radio board. Also, the Universal Software Radio Peripheral (USRP) SDR used in [4, 3] has a maximum bandwidth of 160 MHz. Hence, the actual data rates this approach can achieve are significantly lower compared to 802.11ad. While it allows for in-depth lower layer analysis, the capabilities of the 60 GHz up- and down-converters play a critical role. Similarly to the first approach, the converters typically use horn antennas and mechanical beam steering. In contrast, 802.11ad hardware uses electronically steerable phased-antenna arrays, which can perform a much faster beam search. Hence, such converters allow only for a very limited analysis of the beam search impact. Further, the phase noise of the converters often severely hinders the communication. If this noise is excessive [14], the SDR at the receiver can only measure SNR values, thus forcing researchers to resort to the aforementioned receive sensitivity table in the standard. While manufacturers offer converters which may be suitable in terms of phase noise, they are often custom-built and thus prohibitively expensive. Affordable off-the-shelf devices with low phase noise are hard to find, limiting the suitability of this approach.

2.2.3 Wideband SDRs

Third, full bandwidth setups such as the NI solution [8] mentioned in Section 1 are still very recent. At the time of writing, no research group has used such a system yet for 60 GHz *network* experimentation. While this approach features 2 GHz of bandwidth and is highly configurable, it focuses on the physical layer. To process this large amount of bandwidth in real-time, it uses parallel programming on multiple Field Programmable Gate Arrays (FPGAs). Such a setup may hinder rapid prototyping due to the involved programming complexity. Most importantly, the up- and down-converters are designed for horn antennas only. As discussed above, this only allows for a limited analysis of beam search performance, which is critical in 60 GHz net-

works. At present, electronically steerable phased-array antennas are only available either as custom-built products or as part of COTS consumer hardware, hindering SDR-based wireless networking experiments.

2.2.4 COTS Hardware

Finally, a small number of COTS consumer devices operating in the 60 GHz band are already available. As discussed above, one of the main advantages of such devices is that they typically include a phased-antenna array. This is highly interesting for network experimentation since such arrays suffer from side-lobes, thus increasing interference in networks with a significant number of nodes. As a result, related work often resorts to such devices for practical 60 GHz experiments [18, 9, 15]. Still, 60 GHz COTS devices are often a black box and thus only allow for insights at the upper layers, such as iperf [1] throughput measurements. Early work in this area uses WiHD transmitters to emulate a 60 GHz link [15]. Recently, the Dell D5000 docking station system has become a popular platform for this type of experiments [18, 9]. In a nutshell, the D5000 is a wireless docking station—instead of physically plugging the laptop to the dock, both devices connect via a 60 GHz link. This happens transparently to the user, who interacts with devices attached to the dock as if they were plugged to the laptop. At the physical layer, the dock and the laptop use beamforming and communicate using a protocol based on 802.11ad [9]. The D5000 displays the current physical layer rate in the Graphical User Interface (GUI) available to the user, which allows researchers to deduce the current Modulation and Coding Scheme (MCS). Still, this value is not updated in real-time, thus limiting the insights that it provides, particularly in non-static scenarios.

3. D5000 LOWER LAYER ACCESS

At the time of writing, the Dell D5000 (c.f. Section 2.2.4) is the only commercially available 802.11ad-based device which allows researchers to perform experiments using the full protocol stack. In this section, we first sketch its design and operation. After that, we explain how to enable its monitoring capabilities, and how to obtain lower layer parameters.

3.1 Design and Operation of the D5000

The Dell wireless docking system is based on the Wilocity `Wil6100` 802.11ad solution, also known as "Marlon". In particular, the wireless Network Interface Card (NIC) is divided into the baseband chip `Wil6120` and the radio frequency chip `Wil6110`. The latter includes the phased-antenna array, and is often physically separated from the former, such that the antenna can be placed close to the casing of the device. The cable that connects both chips carries control data, a local oscillator, and the transmit signal at intermediate frequency. The antenna features an 8×2 patch array, and uses a fixed codebook of beampatterns [9].

The dock is designed for Microsoft Windows. A potentially compatible device driver exists for Linux. Still, this driver is designed for a newer version of the Wilocity 802.11ad solution, namely, `Wil6200`. Most importantly, it requires a firmware which is not publicly available[2]. While using the Wilocity card under Linux is highly promising as more lower layer information may be available, the lack of the firmware renders this approach currently unsuitable for network experimentation. Still, this situation may improve with the release of new 802.11ad hardware (c.f. Section 4.4.2).

[2]For more details on this limitation, see https://wireless.wiki.kernel.org/en/users/drivers/wil6210#firmware.

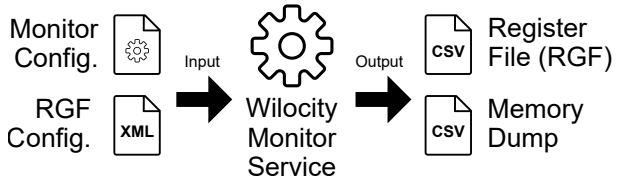


Figure 1: Overview of the Wilocity Monitor.

3.2 Wilocity Monitor

Apparently, the Dell drivers do not provide any access to lower layer information except for the physical layer rate, as explained in Section 2.2.4. However, the driver ships with a so-called "Wilocity Monitor", which runs continuously as a Windows service. This monitor is clearly not meant for usage by the end user, and seems to be a residual of debugging activities. It is configurable via a number of readily accessible text files. Moreover, it stores the monitored physical and MAC layer parameters to a file of comma-separated values (CSV) at configurable intervals. Figure 1 shows an overview of the monitor and the involved files.

3.2.1 Monitor Operation

As shown in Figure 1, the monitor takes two files as input. First, the monitor configuration file specifies the interval at which lower layer parameters of the Wilocity card shall be logged. The minimum interval is 1 ms, which provides a much higher time resolution than the information shown on the device driver GUI. Among others, this file also specifies for how many days logs shall be stored before deletion, and whether the monitor shall perform a memory dump when the service is started. Second, the Register File (RGF) configuration file specifies which lower layer parameters the monitor shall log. The monitor reads any value out of a 712 KB memory area of the Wilocity card. To monitor a value in the memory, the RGF configuration must specify (a) the memory address and (b) the bit mask needed to extract the value from the memory location. The mask selects specific bits out of the memory contents.

Based on the above configuration, the Wilocity Monitor generates two output files. First, it stores a memory dump of the full 712 KB memory at startup. Second, it logs the monitored values in the RGF file at the specified monitoring interval (c.f. the Appendix). These output files are located in a hidden system folder of the operating system.

3.2.2 Lower Layer Parameters

While the Wilocity Monitor provides access to a large memory area containing all kinds of lower layer information, the key problem is to know which combination of memory address and bitwise mask contains which information. From the configuration files of the monitor, it becomes evident that the design of the monitor assumes an additional CSV file which contains a mapping of the memory to human readable mnemonics. Unsurprisingly, this file is not present among the files that ship with the device driver.

Table 1: Monitored Lower Layer Parameters

Description	*Mask Size*
INA Counter Control Packets	32 bits
INA Counter Data Packets	32 bits
CRC OK Counter Control Packets	32 bits
CRC OK Counter Data Packets	32 bits
Baseband Chip Temperature	32 bits
Radio Chip Temperature	32 bits
Automatic Gain Control Attenuation	6 bits
Modulation and Coding Scheme	5 bits
Transmit Beam Pattern	6 bits
Receive Beam Pattern	6 bits

However, the RGF configuration file contains a number of pre-configured values which, once again, seem to be a residual of debugging activities. In Table 1 we show some of these values. The "INA" counters count the number of packets that the physical layer successfully detects based on the frame preamble. Further, the "CRC OK" counters show how many of the detected packets passed the Cyclic Redundancy Check (CRC). We also obtain temperature information of both chips, which may hint at energy consumption. The Automatic Gain Control (AGC) of the receiver is indicated as an attenuation, that is, the higher the logged value, the lower is the gain at the receiver. Finally, Table 1 includes two values which we have inferred based on measurements, namely, the MCS and the transmit/receive beampatterns. In Section 3.3, we explain the methodology which we have developed to determine those values, which may be useful to infer more values in the future.

3.3 Inferring Lower Layer Parameters

Unfortunately, the lower layer parameters originally included in the RGF configuration file neither included the MCS nor details regarding the selected beampatterns. Still, both parameters are highly relevant for 60 GHz network experiments. To infer their memory address and bit mask, we record the memory dump in Figure 1 in a number of controlled environments, and compare them to identify potential memory locations.

Modulation and Coding Scheme. To infer the MCS, we place the laptop and the dock at different distances to induce different MCS values. For each distance, we record the memory dump and the MCS reported in the driver GUI. We then search exhaustively in each memory dump for the corresponding MCS. For each dump, this reveals a number of potential memory locations. To determine the actual location, we compute the intersection of potential locations of all dumps. We repeat the process adding more dumps until a single memory location remains out of the intersection.

Beam Patterns. For the beam patterns, we follow a strategy similar to the one above. Instead of increasing the distance, we change the relative angle of the dock to the laptop. However, in this case the GUI does not reveal any information regarding the current beam pattern which we could use to identify the memory location. Hence, we record multiple memory dumps for each dock and laptop angle. Since the angle does not change, the beam patterns are the same for each dump. We identify all memory positions that do not change for the same angle. Next, we compare which of those memory positions change for different angles. Again, we add angles until one memory location remains.

4. PRACTICAL RESULTS

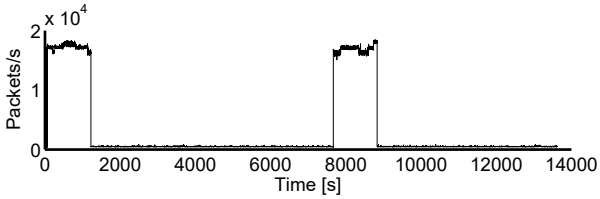
In the following, we analyze the behavior of the parameters in Table 1 for a number of scenarios. First, we study different link distances in a static environment. After that, we consider a scenario with mobility, where the laptop either approaches or departs from the dock. Finally, we analyze the impact of transient link blockage, that is, when an obstacle crosses the 60 GHz link.

4.1 Static Analysis

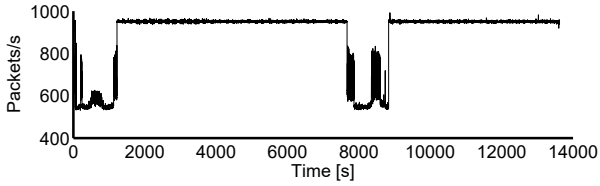
In our first experiment, we place the laptop and the docking station at a fixed distance. We then monitor the parameters in Table 1 while running iperf.

4.1.1 Fixed Link Length

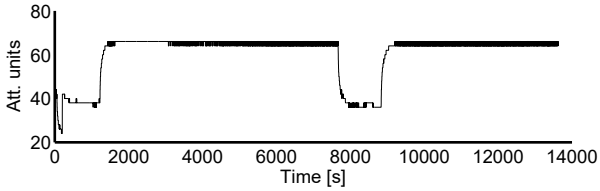
In Figure 2, we show the evolution of lower layer parameters for a total of four hours in a static environment. During the experiment, we start iperf twice for about 20 minutes. As expected, the received packets per second increase to a high value while iperf transmits, and are close to zero otherwise, as shown in Figure 2a. Interestingly, Figure 2b shows that the number of received control packets per second decreases during those periods. The underlying reason is that the Wilocity radio transmits frequent beacons to maintain



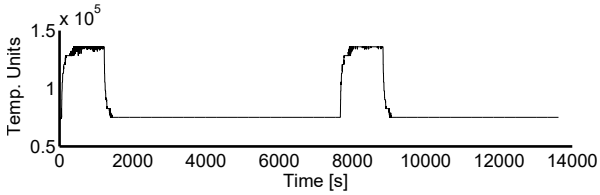
(a) Received physical layer data packets



(b) Received physical layer control packets

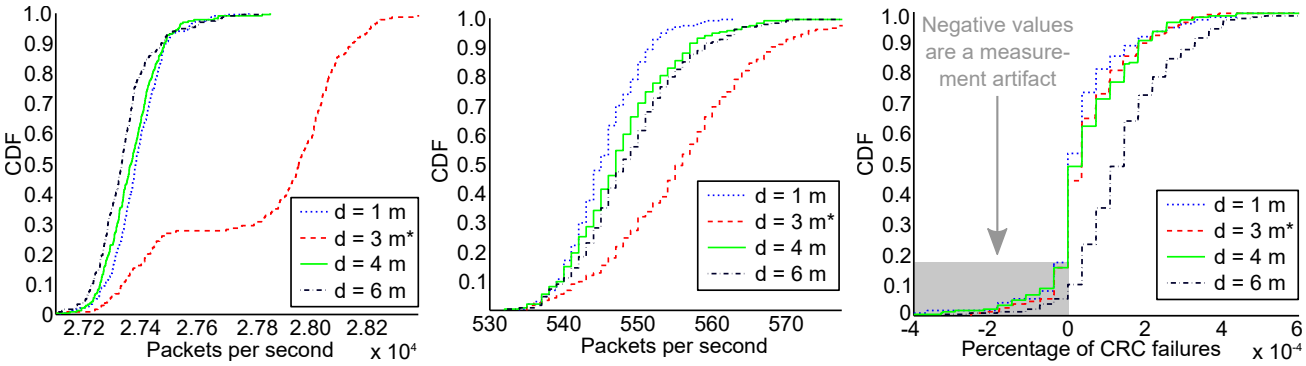


(c) AGC attenuation



(d) Radio chip temperature

Figure 2: Static analysis for fixed link length.



(a) Detected data packets per second. (b) Detected control packets per second. (c) CRC failure rate.

Figure 3: Distance impact. The case for d = 3 meters (marked with *) involved two different MCSs.

connectivity while the link is idle [9], which is unnecessary when the link is active anyhow. The AGC attenuation in Figure 2c is also closely related to the data transmission—while no data is being transmitter, the AGC attenuation increases since no signal amplification is needed. The AGC value that we show in Figure 2c is only related to directional data transmissions, that is, it does not take into account omni-directional control traffic. Also, the monitor logs a dimensionless value, which is hard to relate to an actual attenuation value in dB. Thus, we show the measurement result in "attenuation units". The same is valid for the temperature of the radio chip in Figure 2d. Still, we clearly observe how the temperature increases during data transmissions. This matches our testbed experience, since both the laptop and the dock become particularly warm in the area where the antenna array is located. This suggests that also the energy consumption of the 802.11ad chip modules increases when transmitting data, as expected.

4.1.2 *Distance Impact*

Next, we study the impact of distance on lower layer performance. Specifically, we place laptop and dock at different distances, and analyze the number of detected packets as well as the number of packets that pass the CRC check. The former is always larger than the latter since all packets that pass the check must have been detected before. Hence, we obtain the number of packets that fail the CRC check as the difference of both values. This gives us insights into how distance affects the Packet Error Rate (PER). However, when computing the packets per time unit for detection and CRC decoding, it may happen that the card has computed more CRC checks than the amount of new frames that have been detected during that time unit. In this case, the calculated packet error rate may be negative. However, this measurement artifact is rare, as Figure 3c depicts.

In Figure 3a, we show the Cumulative Distribution Function (CDF) of the packets per second successfully detected at the receiver. We perform the experiment for four different distances, namely, $d \in [1, 3, 4, 6]$ meters. Except for $d = 3$ meters, in this experiment the card uses the same MCS for all distances. We choose this particular case because it allows us to perform a fair comparison of the the CRC failure rate below. Since the MCS is the same, the number of received packets per second is similar. In contrast, for $d = 3$, meters the link switches to a lower MCS halfway through the experiment. As a result, the CDF for $d = 3$ in Figure 3a shows two clearly different areas—as soon as the MCS drops, the number of data packets per second increases since each packet carries less data. This translates into a higher number of control packets in Figure 3b since for $d = 3$ the card transmits more physical layer acknowledgments compared to the other values of d. For those other values of d, we also observe that the number of control packets per second increases slightly with distance. This matches our results in Section 4.2.1, which show that the cards exchange more control traffic to cope with worse link qualities.

Figure 3c provides insights into how the PER increases with distance based on the CRC failure rate. Naturally, this does not include packets that could not be detected since we only consider the statistics at the receiver. Still, the impact of this on the PER should be small since we perform the experiments in a controlled static scenario with no interference. As expected, in Figure 3c we observe that the CRC failure rate increases with distance. While for $d < 4$ meters the CRC failure rate is zero half of the time, for $d = 6$ meters we only achieve zero errors roughly 10% of the time. Still, the maximum failure rate is less than 10^{-3}. As discussed earlier, negative values are a measurement artifact.

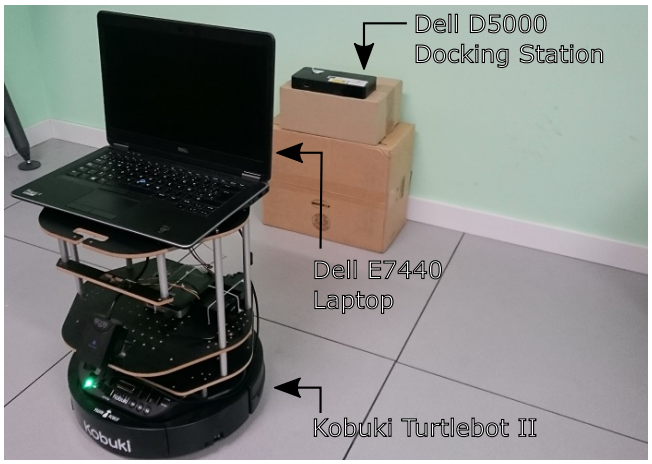
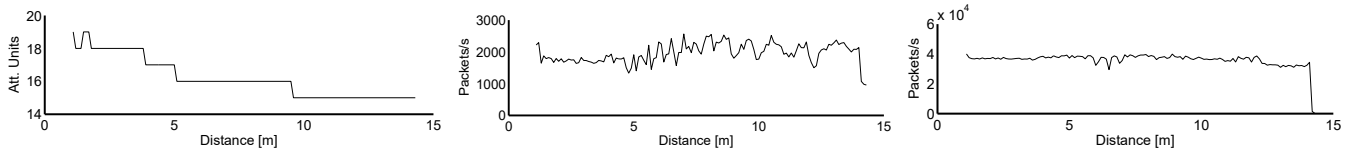
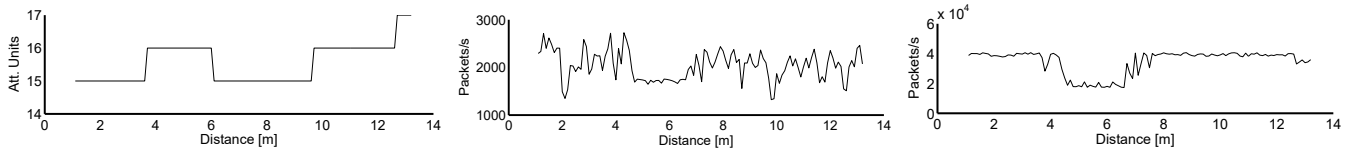


Figure 4: Mobility analysis setup. Wired cables at the docking station are not shown for simplicity.



(a) AGC evolution as link length increases (b) Received control packets per second (c) Received data packets per second

Figure 5: Mobility analysis for a continuously increasing link length.



(a) AGC evolution as link length decreases (b) Received control packets per second (c) Received data packets per second

Figure 6: Mobility analysis for a continuously decreasing link length.

4.2 Mobility Analysis

Our second experiment analyses the evolution of physical layer parameters as the laptop moves towards or away from the docking station. The key difference to Section 4.1.2 is that in this case the movement is continuous, while in our previous analysis we changed the link length in discrete steps. Moreover, in Section 4.1.2 we reset the link for each distance. In contrast, in this experiment we let the AGC, the rate control, and the beam steering adapt automatically to the link length increase. To achieve a smooth continuous movement, we place the laptop on a Kobuki Turtlebot II robot as shown in Figure 4 and let it move along a straight line at a speed of 0.1 m/s. The distance values go from one meter to fourteen meters.

4.2.1 Increasing Link Length

First, we study the case when the link length increases, that is, the robot moves away from the docking station. Figure 5 depicts some selected lower layer parameters. We also show the MCS in Figure 7, which evolves as expected. That is, the MCS decreases from MCS 11 (3850 mbps) to MCS 8 (2310 mbps) as the distance increases. Similarly, the AGC attenuation in Figure 5a also decreases for longer link lengths to compensate for the decreasing signal strength. Further, the number of control packets per second increases slightly and oscillates when distance increases, as depicted in Figure 5b. This suggests that the nodes generate additional control traffic to cope with the worsening link quality. In contrast, the packet rate in Figure 5c remains roughly stable. In other words, the link successfully adapts to the increasing distance in terms of AGC and MCS, hence sustaining the packet transmission. Still, we observe that the transport layer throughput reported in iperf decreases. This is expected, since each packet carries less data due to the lower MCS (c.f. Figure 7).

4.2.2 Decreasing Link Length

Second, we analyze how lower layer parameters evolve as link distance decreases. Figure 6 depicts our results. Note that in this case the robot starts at 14 meters, that is, the figures should be read from right to left. Surprisingly, the

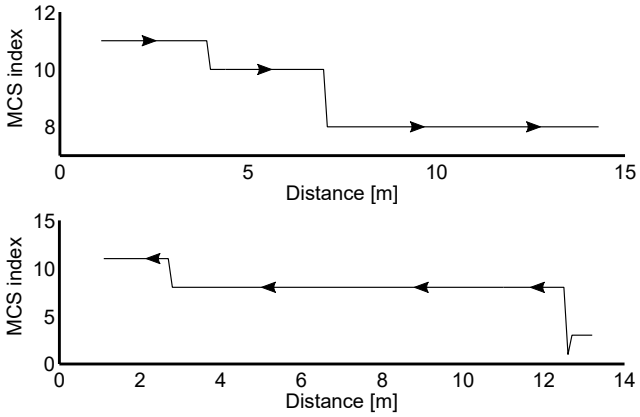


Figure 7: MCS for increasing/decreasing link length

AGC attenuation in Figure 6a decreases as the robot approaches the docking station. We investigated this in detail since this behavior is very counter-intuitive. We observed that the AGC of the Wilocity cards tends to behave in a conservative manner. That is, while no transmission is ongoing, it remains at a high attenuation value to prevent clipping in case of a strong incoming signal. When a transmission starts, such as at the beginning of our experiment in Figure 6a, the attenuation decreases to maximize the signal at the input of the Analog to Digital Converter (ADC). However, in Figures 6b and 6c we observe that the number of successfully received control and data packets clearly decreases at roughly seven meters to the docking station. This suggests that clipping occurred due to the increase in signal strength. Shortly after this, the AGC reacts by increasing again the attenuation. The results show that, after that, the number of received packets per second recovers. Finally, the AGC attenuation decreases again but no clipping seems to occur. This may be due to, for instance, the dock entering an area for which the transmit beampattern had a lower gain. Finally, we also observe that, as expected, the MCS in Figure 7 increases as the link length decreases—starting at MCS 3 (962.5 mbps), it reaches up to MCS 11 (3850 mbps).

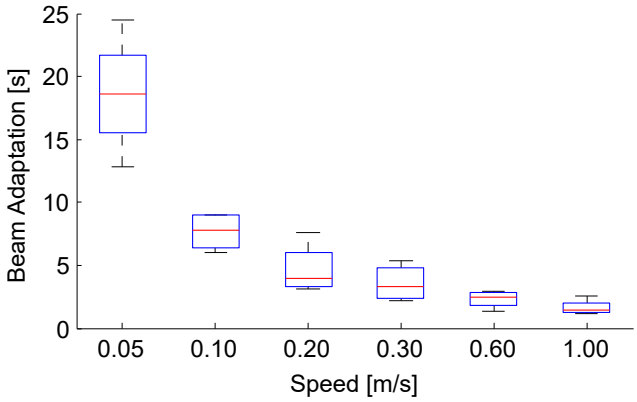


Figure 8: Duration of beampattern adaptation.

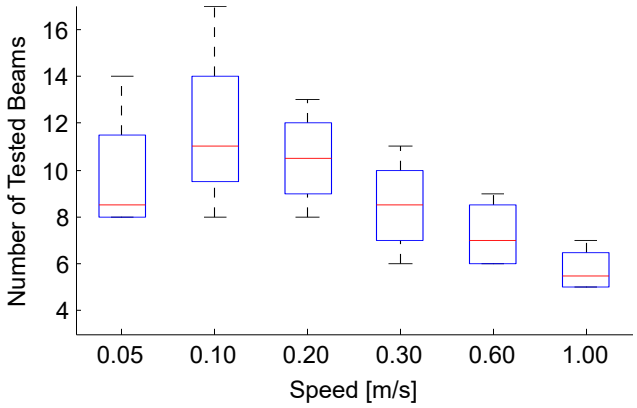


Figure 9: Number of beam tests during blockage

4.3 Transient Blockage Analysis

In our last experiment, we consider a transient blockage scenario. To this end, we place the laptop and the dock at a distance of three meters and interrupt the link with a continuously moving obstacle. That is, the obstacle enters and leaves the Line-Of-Sight (LOS) path without stopping at any time. Specifically, we use the Kobuki Turtlebot II as a moving obstacle. We repeat the experiment for multiple speeds to study the impact of the blockage duration on how the nodes adjust their beampatterns to cope with blockage.

As soon as the robot enters the LOS path, the SNR at the receiver drops. As a result, the nodes test different transmit and receive beampattern combinations to try to avoid the obstacle. Since the obstacle moves continuously, the beam adaptation process is typically in vain—a working beampattern combination quickly becomes sub-optimal as the robot continues its movement. Figure 8 shows the duration of this process. As expected, the process is shorter the faster the robot moves since it spends less time blocking the LOS path. However, the length of the beam adaptation process is not necessarily related to the number of beampattern combinations that the nodes test, as shown in Figure 9. In particular, when the robot moves at 0.05 m/s, the nodes test less beampatterns than at 0.1 m/s. This is likely because the SNR variations as the robot moves through the link are smoother, and thus the beam steering algorithm sets in less often. As speed increases, initially we observe the aforementioned increase in the number of tested beampatterns but then again a decrease. The latter is expected since the nodes simply have less time to test beams as the blockage becomes shorter. We also analyzed whether the beampatterns before and after the blockage match. Interestingly, this only happened in roughly 50% of our experiments, with no clear relation to the robot speed. This shows that a transient link blockage may induce a long-term sub-optimal beam alignment on a 60 GHz link, thus having a significant impact on performance. We discuss this issue further in [7].

4.4 Discussion

4.4.1 *Testbed Capabilities*

Our practical results in this section show that our lower layer monitoring approach can provide detailed insights into the operation of 60 GHz networks at a fraction of the cost than any of the approaches surveyed in Section 2. Most interestingly, with many of those approaches gaining the above insights is unfeasible or requires a high implementation overhead. For instance, obtaining an estimate of the PER for a wideband 60 GHz system taking into account the impact of realistic beampatterns and beam steering behaviors would require a significant effort on other platforms but is simple in our case. Such PER traces can be used for further evaluation tools, such as 60 GHz network simulators, which to date typically use very basic physical layer models. All in all, we conclude that our approach turns what originally was a black box into a highly useful tool.

4.4.2 *Future Work*

In future work, we plan to infer the content of more memory addresses of the Wilocity cards. Moreover, upcoming COTS 60 GHz devices build on an evolution of the cards in our devices. Initial investigations have revealed that the new cards offer similar monitoring capabilities than our cards. While the "Wilocity Monitor" is probably not available on new devices, the firmware offers an equivalent interface to obtain the same lower layer information. Most interestingly, the new devices are allegedly supported under Linux using the "wil6210" [2] driver by Qualcomm. This opens the door to a wide range of experiments, providing an excellent platform for 60 GHz networking research.

A highly interesting extension to our work is to not only retrieve lower layer information but to also modify it. This would allow us to, for instance, set the MCS or the beampattern in use to a specific value. As a result, researchers would be able to implement and evaluate their own rate or beam adaptation algorithms on COTS hardware. Unfortunately, the interface that we use to access the card memory is read only. Moreover, the firmware on the card directly controls the lower layer operation, overriding any external commands. In other words, controlling the lower layer behavior of COTS 60 GHz devices remains an open challenge.

5. CONCLUSION

In this paper, we explain how to use hidden monitoring capabilities in a 60 GHz COTS device for research purposes. This provides access to lower layer information, thus enabling an in-depth analysis of the performance of 60 GHz networks. In contrast to existing 60 GHz experimentation

platforms, our approach is affordable and takes into account the impact of the complete networking stack. Specifically, we show how to monitor parameters such as the decoding PER, the AGC, the MCS, and the beampatterns that nodes use. We validate the monitoring capabilities of our approach in practice by means of extensive testbed measurements.

6. ACKNOWLEDGMENTS

This work is partially supported by the European Research Council grant ERC CoG 617721, the Ramon y Cajal grant from the Spanish Ministry of Economy and Competitiveness RYC-2012-10788, and the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919). We also thank our interns Alba De Juanas Sanz and Alejandro Ranchal Pedrosa for helping us to take some of the measurements shown in this paper.

7. REFERENCES

[1] Iperf - The TCP/UDP Bandwidth Measurement Tool. http://dast.nlanr.net/Projects/Iperf/. [Accessed June 3, 2016].

[2] Linux Wireless wil6210 Documentation. https://wireless.wiki.kernel.org/en/users/drivers/wil6210. [Accessed August 5, 2016].

[3] J. Ansari, N. Perpinias, A. Nähring, P. Mähönen, and M. Petrova. Empirical Characterization of mm-Wave Communication Links in Realistic Indoor Scenarios. In *Proc. of WCNC'15*, 2015.

[4] J. Arnold, L. Simic, M. Petrova, and P. Mähönen. Demo: Spectrum-Agile mm-Wave Packet Radio Implementation on USRPs. In *Proc. of SRIF'15*, 2015.

[5] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band. *IEEE Std 802.11ad-2012*, pages 1–628, Dec 2012.

[6] LeTV. Le Max Pro Announcement. http://www.tomshardware.com/news/letv-le-max-pro-announcement,31028.html. [Accessed June 3, 2016].

[7] A. Loch, I. Tejado, and J. Widmer. Potholes Ahead: Impact of Transient Link Blockage on Beam Steering in Practical mm-Wave Systems. In *Proc. of European Wireless Conference*, 2016.

[8] National Instruments. mmWave Transceiver System. http://www.ni.com/sdr/mmwave/. [Accessed June 3, 2016].

[9] T. Nitsche, G. Bielsa, I. Tejado, A. Loch, and J. Widmer. Boon and Bane of 60 GHz Networks: Practical Insights into Beamforming, Interference, and Frame Level Operation. In *Proc. of CoNEXT'15*, 2015.

[10] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer. Steering with Eyes Closed: mm-Wave Beam Steering Without In-Band Measurement. In *Proc. of IEEE INFOCOM'15*, 2015.

[11] T. S. Rappaport, S. Sun, R. Mayzus, H. 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.

[12] S. K. Saha, V. V. Vira, A. Garg, and D. Koutsonikolas. 60 GHz Multi-Gigabit Indoor WLANs: Dream or Reality? *CoRR*, abs/1509.04274, 2015.

[13] L. Simić, N. Perpinias, and M. Petrova. 60 GHz Outdoor Urban Measurement Study of the Feasibility of Multi-Gbps mm-Wave Cellular Networks. In *Proc. of mmNet'16, co-located with IEEE INFOCOM'16*, 2016.

[14] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan. 60 GHz Indoor Networking Through Flexible Beams: A Link-Level Profiling. In *Proc. of ACM SIGMETRICS'15*, 2015.

[15] X. Tie, K. Ramachandran, and R. Mahindra. On 60 GHz Wireless Link Performance in Indoor Environments. In *Proc. of PAM'12*, 2012.

[16] TP-Link. Talon AD7200 Multi-Band Router. http://www.tp-link.com/en/products/details/AD7200.html. [Accessed June 3, 2016].

[17] H. Xu, V. Kukshya, and T. S. Rappaport. Spatial and temporal characteristics of 60-ghz indoor channels. *IEEE Journal on Selected Areas in Communications*, 20(3), 2002.

[18] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao, and H. Zheng. Demystifying 60GHz Outdoor Picocells. In *ACM MobiCom'14*, 2014.

APPENDIX

The monitor configuration file in Figure 1 specifies the interval at which the "Wilocity Monitor" shall log the lower layer information specified in the RGF configuration file. The interval is specified as an integer in milliseconds, which suggests a minimum monitoring interval of one millisecond. However, if the interval is very short, the actual value may differ from the expected behavior. The interval is limited due to hardware issues in terms of, for instance, the speed of the hard drive of the machine hosting the Wilocity card. For a Dell E7440 laptop with a Solid-State Drive, we obtained the performance shown in Table 2. For intervals larger than 100 ms, the monitor records as many logs as expected. However, at shorter intervals, the performance starts diverging from the expected behavior. The maximum number of log entries per second is roughly 60 logs/s, which corresponds to an interval of about 16 ms.

Interestingly, for intervals between 100 ms and 16 ms, the behavior is not consistent. For instance, for an interval of 50 ms we only obtain about 15 logs/s instead of 20 logs/s even though the device can record up to 60 logs/s. This demonstrates that the performance of the monitor also depends on more factors, such as the internals of the operating system running the monitor service. For experimentation using the monitor, we recommend an interval of 100 ms or larger.

Table 2: Recorded Log Entries Per Second

Interval n	*Theoretical speed*	*Actual speed*
$n \geq 100$	$1000/n$ logs/s	$1000/n$ logs/s
50	20 logs/s	15 to 16 logs/s
25	40 logs/s	30 to 31 logs/s
20	50 logs/s	30 to 31 logs/s
10	100 logs/s	59 to 63 logs/s
5	200 logs/s	59 to 63 logs/s
2	500 logs/s	59 to 63 logs/s
1	1000 logs/s	59 to 63 logs/s