Enabling High-Quality Untethered Virtual Reality

by

Austin Duffield

S.B., Massachusetts Institute of Technology (2016)

Submitted to the Department of Electrical Engineering and Computer
Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2017

Author . . . .
Department of Electrical Engineering and Computer Science
August 4, 2017

Certified by. . . . .
Dina Katabi
Professor
Thesis Supervisor

Accepted by . . . .
Christopher J. Terman
Chairman, Master of Engineering Thesis Committee

Enabling High-Quality Untethered Virtual Reality

by

Austin Duffield

Abstract

Today's virtual reality (VR) headsets require a cable connection to a PC or game console. This cable significantly limits the player's mobility and, hence, her VR experience. The high data rate requirement of this link (multiple Gbps) precludes its replacement by WiFi. Thus, in this thesis, we focus on using mmWave technology to deliver multi-Gbps wireless communication between VR headsets and their game consoles. We address the two key problems that prevent existing mmWave links from being used in VR systems. First, mmWave signals suffer from a blockage problem, i.e., they operate mainly in line-of-sight and can be blocked by simple obstacles such as the player lifting her hand in front of the headset. Second, mmWave radios use highly directional antennas with very narrow beams; they work only when the transmitter's beam is aligned with the receiver's beam. Any small movement of the headset can break the alignment and stall the data stream. We present MoVR, a novel system that allows mmWave links to sustain high data rates even in the presence of a blockage and mobility. MoVR does this by introducing a smart mmWave mirror and leveraging VR headset tracking information. We implement MoVR and empirically demonstrate its performance using an HTC VR headset.

Thesis Supervisor: Dina Katabi
Title: Professor

Disclaimer

This project is joint work with graduate student Omid Abari and postdoctoral researcher Dinesh Bharadia. Much of the system presented in this thesis was developed during joint brainstorming sessions. I was responsible for designing the MoVR controller, as well as implimentation and empirical evaluation of the system. I really enjoyed working with Omid and Dinesh and appreciate all the help.

Acknowledgments

Contents

List of Figures

Chapter 1

Introduction

The past few years have witnessed major advances in augmented reality and virtual reality (VR) systems, which have led to accelerated market growth. Facebook has recently started shipping their VR headset (Oculus Rift) and expects to ship more than 2 million headsets by 2017 [6]. HTC sold more than 15,000 VR headsets in the first 10 minutes following their release [8]. These devices are expected to soon dominate the gaming and entertainment industry, and they have found applications in manufacturing and healthcare [20, 21]. However, a key challenge prevents this technology from achieving its full potential. High-quality VR systems need to stream multiple Gbps of data from their data source (PC or game console) to the headset. As a result, these headsets have an HDMI cable snaking down the player's neck and hardwiring her to the PC, as shown in Fig 1-1. The cable not only limits the player's mobility and interferes with the VR experience, but also creates a tripping hazard since the headset covers the player's eyes. This has left the industry searching for untethered solutions that can deliver a high-quality VR experience without these limitations. Unfortunately, typical wireless systems, such as WiFi, cannot support the required data rates. This challenge has led to awkward products: Zotac has gone as far as stuffing a full PC in the player's backpack in the hope of delivering an untethered VR.

Ideally, one would like to replace the HDMI cable with a wireless link. Thus, multiple companies have advocated the use of mmWave for VR since mmWave radios

Figure 1-1: **Virtual Reality experience:** The headset's cable not only limits the player's mobility but also creates a tripping hazard.

have been specifically designed to deliver multi-Gbps data rates [16, 1]. The term mmWave refers to high frequency RF signals in the range of 24 GHz and higher [15, 7]. The 802.11ad standard operates in mmWave and can transmit over 2 GHz of bandwidth and deliver up to 6.8 Gbps. No other consumer RF technology can deliver such data rates. However, mmWave links bring up new challenges that must be addressed before this technology can be used for VR applications:

- **Dealing with blockage:** mmWave links require a line-of-sight between transmitter and receiver, and they do not work well through obstacles or reflections. This problem is due to the fact that mmWave antennas are highly directional and typically generate narrow beams. Hence, even a small obstacle like the player's hand can block the signal. Said differently, these links work well when the receiver on the headset has a clear line-of-sight to the transmitter connected to the PC, but if the player moves her hand in front of the headset (see Fig. 4-1), or other people in the environment obstruct the receiver's view to the transmitter, the signal will be temporarily lost, causing a glitch in the data stream (shown in our empirical results in section 4). While temporary outages are common in wireless communication, VR data is non-elastic, and cannot tolerate any degradation in SNR and data rate.

14

- **Dealing with mobility:** Since mmWave radios use highly directional antennas, they work only when the transmitter's beam is aligned with the receiver's beam. Further, since the wavelength is very small, even a small movement of the headset can hamper the alignment and break the link. Past work on mmWave typically assumes static links and, hence, fixed alignment [39, 48, 32]. Identifying the correct alignment for the antennas can take up to multiple seconds [49, 44]. Such delay is unacceptable for VR systems, which need to play a new frame every 10 milliseconds, even when the headset moves [10].

This thesis introduces MoVR, a wireless system that enables a reliable and high-quality untethered VR experience via mmWave links. MoVR addresses the main challenges facing existing mmWave links. In particular, MoVR overcomes the blockage problem by introducing a self-configurable mmWave mirror that detects the incoming signal and reconfigures itself to reflect it toward the receiver on the headset. In contrast to a traditional mirror, a MoVR mirror does not require the angle of reflection to be equal to the angle of incidence. Both angles can be programmed so that our mirror can receive the signal from the mmWave transmitter attached to the data source and reflect it towards the player's headset, regardless of its direction. In section 4.2, we explain the design of such mmWave mirrors and how they can be implemented simply by deflecting the analog signal without any decoding.

Next, MoVR ensures that the VR system sustains high data rates to the headset in the presence of mobility. In contrast to past work on mmWave [1, 26, 49], MoVR does not scan the space to find the best way to align the mmWave directional antennas, a process known to incur significant delay [49, 44]. Specifically, MoVR finds the best beam alignment by relying on existing tracking functions available in VR systems. In designing MoVR, we observe that VR systems already track the location of the headset to update the 3D view of the player. Thus, MoVR leverages this information to quickly localize the headset and move the transmitter antenna's beam with it. However, while the VR application tracks the movements of the headset, it does not know the location of the headset with respect to the mmWave transmitter and the MoVR mirror. Thus, we design a novel algorithm that combines the output of VR

headset tracking with RF-based localization to quickly steer the mmWave antennas and keep their beams aligned as the player moves around.

We have built a prototype of MoVR and evaluated its performance empirically using an HTC VR system. Our results can be summarized as follows:

- In the absence of MoVR's mirror, even a small obstacle like the player's hand can block the mmWave signal and result in a drop in SNR of 20dB, leaving the VR headset with no connectivity. The addition of MoVR's mirror prevents the loss of SNR in the presence of blockage, sustaining high data rates.

- Given the VR headset information, MoVR aligns the antenna beams in under a few micro seconds, which is negligible compared to the user's movement. Further, the resulting alignment sustains the required high SNR and VR data rates.

- Finally, in a representative VR gaming setup, MoVR provides an SNR of 24dB or more for all locations in the room and all orientations of the headset, even in the presence of blockage and player mobility. This SNR is much higher than the 20dB needed for the VR application.

Chapter 2

Related Work

Related work can be classified into three areas.

(a) **Virtual Reality:** Existing VR systems can be divided into PC-based VR, like Occulus Rift and HTC Vive, and Gear VR, like systems by Samsung and Visus [17, 22]. PC-based VR systems leverage their computational horsepower to generate rich graphics that look realistic and support fast head motion. They require, however, an HDMI cable to connect the PC to the headset. Gear VR slides a powerful smart phone into the headset. Thus, they do not need an external cable. Their mobility, however, is limited by the inability to support rich graphics that react to motion; their imagery tends to blur with motion [3]. There is a huge interest in untethered PC-based VR systems. Optoma and SiBeam have proposed using mmWave radios to connect the headset to the PC, but they have not provided any details about their proposal [16, 1]. Sulon proposed to equip the headset with an integrated computer [18]. Unfortunately, this would make the headset much larger and heavier, thus interfering with the user experience. WorldViz advertises a wireless wide-area tracking system. However, they still require the user to have a cable for the display or carry a limited data source and a processor unit [25]. Zotac advertises a mobile VR headsets where the user carries the PC in a backpack. Finally, Google has recently announced that their next VR headset will be wireless, but has not provided any details of the design or the release date [23].

(b) **mmWave Communications:** Much past work on mmWave communication

addresses *static links*, such as those inside a data center [39, 48, 32], where there is a line-of-sight path between the transmitter and receiver. Some past work looks at mobile links for cellular networks or wireless LANs [46, 43, 40]. These systems typically scan the space to align the antennas, a process that takes up to several seconds [49, 44]. In contrast, by leveraging the fact that VR systems already track the headset, MoVR is able to speed up antenna steering enough that it can be done faster than the VR frame rate, as we show in section 7. Also, most past work on mmWave links assumes line-of-sight connectivity. Some papers do consider scenarios in which the line-of-sight between transmitter and receiver is blocked [46, 45, 38]. However, since they target elastic applications, their solution switches the directional antenna to the best reflected path, which typically has a much lower SNR (see Fig. 4-2). In contrast, the VR application is non-elastic and cannot tolerate reduction in its SNR and data rate. Also, there are wireless HDMI (WHDI) products from LG and Samsung which operate at mmWave frequencies, but these products assume static links and require line-of-sight between the receiver and transmitter [24]. Thus, they cannot adapt their direction and will be disconnected if the player moves. Finally, the work in [48] has proposed a form of mmWave mirror to reflect an RF signal off the ceiling of a data center. Their approach, however, covers the ceiling with metal. Such a design is unsuitable for home applications and cannot deal with player mobility.

(c) **Relay and Full-Duplex:** The design of MoVR mirrors is related to that of wireless relays at lower frequencies (e.g., Wi-Fi and LTE [19]). Similarly to a MoVR mirror, these relays amplify and forward the signal of interest. However, they do not deal with the issue of directionality. In contrast, our MoVR mirror needs to capture the mmWave signal along a particular direction and reflect it in the direction of the headset. Finally, MoVR mirrors are related to previous work on full-duplex relays since they receive a signal and transmit it at the same time. However, full-duplex radios require complex analog and digital hardware with full transmit and receive chains [29]. In contrast, MoVR mirrors have only an analog front-end (i.e., antennas and an amplifier) and do not need digital transmit or receive chains.

Chapter 3

Overview

MoVR is a wireless communication system for virtual reality applications. It enables a sustainable, high-data-rate wireless link even in the presence of blockage and headset mobility. High-quality VR systems stream multiple Gbps from a high-power PC to the headset. MoVR delivers this data over a mmWave wireless link. Fig. 3-1 shows MoVR's setup. The PC is connected to a mmWave transmitter, which we refer to as the AP, and the headset is equipped with a mmWave receiver. As the figure shows, MoVR operates in two modes, depending on the real-time scenario: when the direct path from the AP to the headset is clear, the AP beams its signal to the headset. However, if the direct path is blocked, the AP detects the blocking and reflects its signal to the headset via a MoVR mirror. The environment may have one or more MoVR mirrors; the AP picks the best one depending on the headset location.

The next few sections present the components that contribute to the design of MoVR. We start by explaining the two key challenges in using mmWave links in VR systems, and how we overcome them. We then explain how the various components work together to satisfy the application.
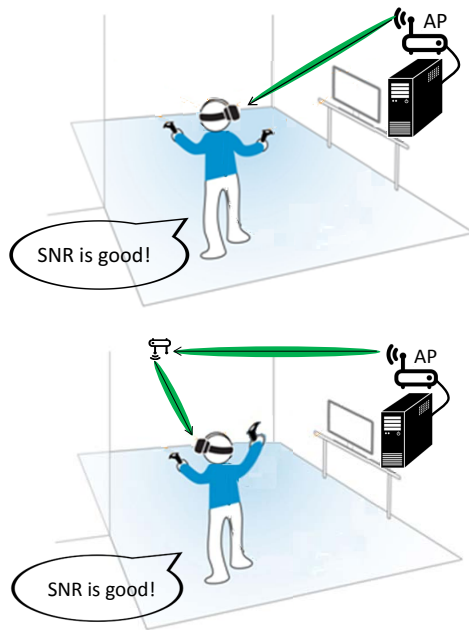
Figure 3-1: **MoVR's setup:** MoVR's setup. The PC is connected to a mmWave AP and the headset is equipped with a mmWave receiver. In the case of a blockage (e.g., the user raises her hand or turns her head), the AP delivers its signal by reflecting it off a MoVR mirror.

Chapter 4

Blockage Problem

A key challenge in using mmWave links for VR applications is that they may be easily blocked by a small obstacle, such as the player's hand. This is a side effect of highly directional antennas, which mmWave radios must use to focus their power and compensate for path loss. Below, we investigate the impact of blockage in more detail, then explain our solution to overcome this problem.

4.0.1 Impact of Blockage

We first investigate the impact of blocking the direct line-of-sight on the signal's SNR and the link's data rate. To do so, we attach a mmWave radio to an HTC PC-based VR system and another one to the headset (see section 7 for hardware details). We conduct experiments in a $5m \times 5m$ room. We place the headset in a random location that has a line-of-sight to the transmitter, and measure the SNR at the headset receiver. We then block the line-of-sight and measure the SNR again. We consider different blocking scenarios: blocking with the player's hand, blocking with the player's head, and blocking by having another person walk between headset and the transmitter. We repeat these measurements for multiple different locations. Fig. 4-2 shows the results of this experiment: the top graph shows the SNR and the bottom graph shows the data rate. The SNRs are measured empirically and the corresponding data rates are computed by substituting the SNR measurements into
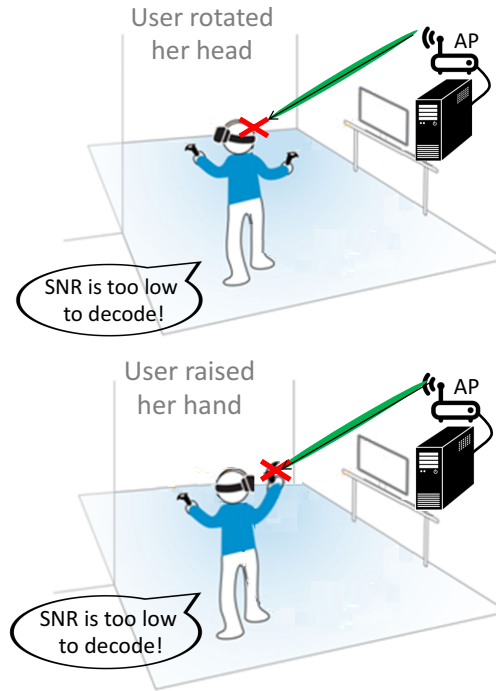
Figure 4-1: **Blockage Scenarios:** As the user moves his head or hand, the line-of-sight path between the AP and the headset's receiver can be easily blocked. This results in a significant drop in SNR and data rate

standard rate tables based on the 802.11ad modulation and code rates [12, 13, 9]. The first bar in Fig. 4-2 shows that, in the absence of blocking, the mean SNR is 25dB and the resulting data rate is almost 7 Gbps, which exceeds the needs of the VR application. Bars 2, 3, and 4 in the figure correspond to different blocking scenarios. They show that even blocking the signal with one's hand degrades the SNR by more than 14 dB and causes the data rate to fail to support the VR application.

One solution to overcome this challenge is to rely on non-line-of-sight paths –i.e., the signal reflections from walls or other objects in the environment. For example, both the transmitter and the headset receiver can direct their signal beams toward a wall and rely on the natural reflection from the wall. In fact, this is how current mmWave systems work. Unfortunately, non-line-of-sight paths typically have much higher attenuation than the line-of-sight path due to the fact that walls are not perfect reflectors and therefore scatter and attenuate the signal significantly. Moreover, signals travel a longer distance in non-line-of sight scenarios than in line-of sight
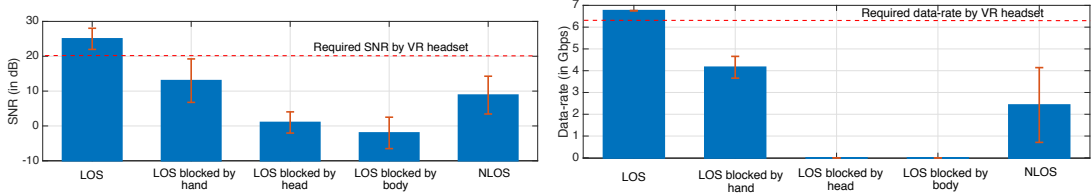
Figure 4-2: **Blockage impact on data rate.** SNR and data rate for different scenarios: line-of-sight (LOS) without any blockage, LOS with different blockages and non-line-of-sight (NLOS). The figure shows that blocking the signal with one's hand, head, or body results in a significant drop in SNR and causes the system to fail to support the required VR data rate. The figure also shows that simply relying on NLOS reflections in the environment does not deliver good SNR and would fail to support the required data rate.

scenarios, which results in higher attenuation.

To confirm, we repeat the measurements for all blocking scenarios, but instead of trying to receive the signal along the blocked direct path, we sweep the mmWave beam on the transmitter and receiver in all directions. We try every combination of beam angle for both transmitter and receiver antennas, with 1 degree increments. We ignore the direction of the line-of-sight and note the maximum SNR across all non-line-of-sight paths. The last bar in Fig. 4-2 shows the results of this experiment. It shows that when the transmitter and receiver have to use a non-line-of-sight path, the SNR drops by 16dB on average. The figure also shows that this reduction in SNR causes the data rate to fail to support the VR application.

Note that one cannot solve the blockage problem by putting more antennas on the back or side of the headset, since the line-of-sight from the AP to the headset may get completely blocked by the player's hands or body (as shown in Fig. 4-1), or by the furniture and other people in the environment. One naïve solution to overcome this challenge is to deploy multiple mmWave APs in the room to guarantee that there is always a line of sight between the transmitter and the headset receiver. Such a solution requires extending many HDMI cables in the environment to connect each AP to the PC. However, this defeats the purpose of a wireless design because it requires enormous cabling complexity. Further, requiring multiple full-fledged mmWave transceivers will significantly increase the cost of VR systems and limit their adoption
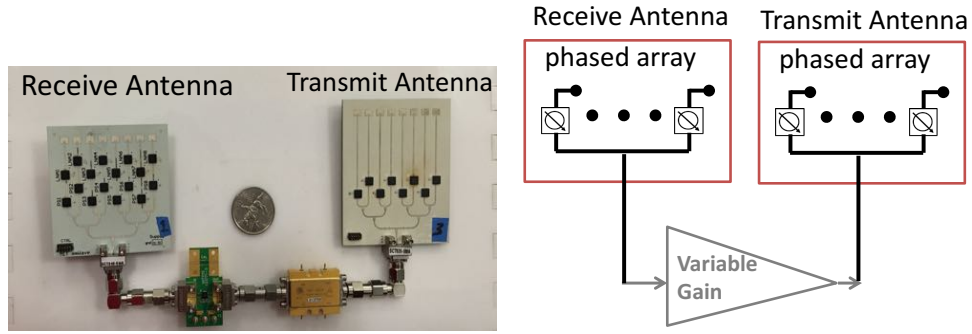
Figure 4-3: **MoVR programmable mmWave mirror:** (a) the implementation and (b) the block diagram of the mirror. The design of the mirror is small and simple. It consists of two directional phased-array antennas connected via a variable-gain amplifier.

in the market.

In the next section, we describe how to overcome the blockage problem using mmWave mirrors. Our mirrors do not need to connect to the PC, and have a simple cheap design with no digital transmit or receive components.

4.0.2 Programmable mmWave Mirrors

To overcome the blockage problem, we designed a programmable mmWave mirror that can control both the angles of incidence and reflection [27]. Fig. 4-3 shows a basic diagram of the circuit and a picture of our prototype. Each MoVR mirror consists of a transmit and receive antenna connected via a variable-gain amplifier. As is common in mmWave radios, the antennas are implemented using phased arrays in order to create highly-directional beams, which can be steered electronically in a few microseconds. Note that the design is quite simple. Specifically, it neither decodes the signal nor includes any transmit or receive digital components (DAC, ADC, mixer, etc.). This allows us to avoid complex and expensive components that would have to operate at multiple Gbps.

An important challenge in designing such a mmWave mirror stems from the leakage between the transmit and receive antennas. At a high level, a MoVR mirror works by capturing the RF signal on its receive antenna, amplifying it, and reflecting it using a transmit antenna. However, some of the signal reflected by the mirror is also
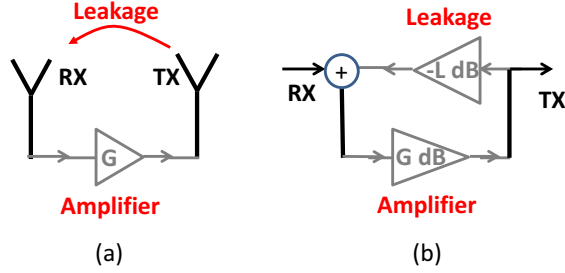
Figure 4-4: **MoVR's mirror block diagram:** (a) the block diagram and (b) equivalent signal-flowgraph of MoVR's mirror. The figure shows that the input signal is first amplified by G_{dB}, then attenuated by L_{dB} and fed back to the input as a leakage.
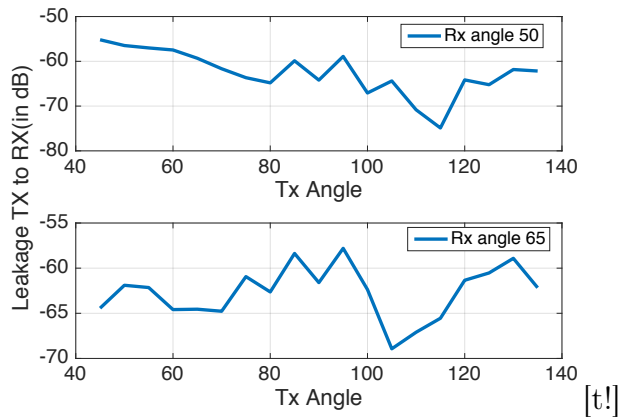
Figure 4-5: **Leakage between mirror's transmit and receive antennas**: The leakage across different transmit beam directions for two different receive beam directions. The figure shows that the leakage variation can be as high as 20dB. This result confirms a need for an adaptive algorithm that reacts to the leakage in real time and adjusts the amplifier gain accordingly.

received by its own receive antenna. This means that the output of the amplifier is fed back to the input of the amplifier. This creates a feedback loop that can cause the amplifier to saturate, thereby generating garbage signals. Thus, a key question in designing MoVR mirrors is: how do we set the optimal amplifier gain so that we avoid saturation, but also maximize the SNR delivered to the headset?

In order to ensure that the leaked signal is damped while the signal of interest (i.e., the received signal from the AP) is amplified, we need to ensure that the amplifier gain is less than the leakage. To see why this is the case, consider the signal-flow-graph of the mirror, shown in Fig. 4-4(b). The input signal is first amplified by G_{dB}, then attenuated by L_{dB} and fed back to the input. From Control Theory [33], we know

25

that for this system to stay stable we need to ensure that $G_{dB} - L_{dB} < 0$ [33, 30]. This implies that the amplifier gain (G_{dB}) must be lower than the absolute value of the leakage (L_{dB}); otherwise the system becomes unstable, leading to saturation of the amplifier.

To avoid this saturation, the mirror needs to measure the leakage and then set the amplification gain lower than the leakage. The leakage, however, varies when the direction of the antenna beam changes to track the headset. Fig. 4-5 shows the leakage across different transmit beam directions for two different receive beam directions. As we can see, the leakage variation can be as high as 20dB. The variation of the leakage and the fact that the amplifier gain must always be set lower than the leakage create a need for an adaptive algorithm that reacts to the leakage in real time and adjusts the amplifier gain accordingly.

One naïve algorithm is to send a signal from the mirror's transmit antenna and measure the received power at its receive antenna in order to estimate the amount of leakage, then to use this information to set the amplifier gain accordingly. However, we cannot do this since a MoVR mirror does not have digital transmit and receive chains.

Our solution exploits a key characteristic of amplifiers: an amplifier draws significantly higher current (from a DC power supply) as it gets close to saturation mode, compared to during normal operation [37, 31].[1] We can therefore detect if the amplifier is getting close to its saturation mode by monitoring the current consumption from the power supply. Thus, our gain control algorithm works as follows: it sets the amplifier gain to the minimum. It increases the amplifier gain step by step while monitoring the amplifier's current consumption. The algorithm continues increasing the gain until the current consumption suddenly goes high. This indicates that the amplifier is entering its saturation mode. The algorithm then backs off, keeping the amplification gain just below this point.

---

[1]The exact quantity of the amplifier's current consumption for its different operating modes are specified in its datasheet. We use a simple IC which measures the current consumption of the amplifier to detect its operating mode.
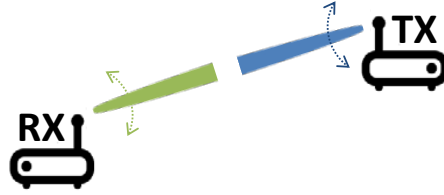
Figure 5-1: **Beam alignment in mmWave radios:** mmWave radios need to find the best alignment between the transmitter's and receiver's beams to establish a communication link.

Chapter 5

Dealing with Mobility

Movement of the VR headset creates a critical challenge for mmWave links. Specifically, mmWave frequencies suffer from a large path loss. To compensate for this loss, mmWave radios use highly directional antennas to focus the signal power in a narrow beam. Such directional antennas can be implemented using phased arrays. In fact, since the wavelength is very small (on the order of a millimeter), tens or hundreds of such antennas can be packed into a small space, creating a pencil-beam antenna. The beam can be steered electronically in a few microseconds. However, the real challenge is to identify the correct spatial direction that aligns the transmitter's beam with the receiver's beam (as shown in Fig. 5-1). This is particularly difficult in VR applications since the headset is naturally in a mobile state.

Below, we investigate the impact of beam misalignment on the signal and explain our solution to overcome this problem.
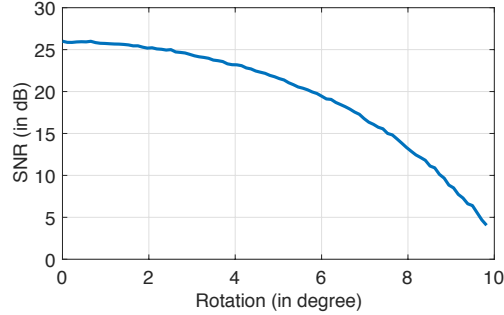
Figure 5-2: **SNR versus amount of headset rotation:** Even a minor head rotation of a few degrees can cause a major degradation in the received SNR. This result confirms the need for real-time beam tracking to realign the transmitter's and the receiver's beams as the player moves her head.

5.0.1   Impact of Beam Misalignment

We first investigate the impact of beam misalignment on the SNR of the signal delivered to the headset. To do so, we attach a mmWave transmitter to the VR PC (which we call the AP), and a mmWave receiver to the headset. We position the headset's receiver such that it has a line-of-sight path to the AP's transmitter. We ensure that the transmitter's and receiver's beams are perfectly aligned by scanning for all possible alignments and picking the one that maximizes the SNR. We use this setup as the initial position of the headset in our experiment –i.e., we start the experiment with a perfect beam alignment. We then rotate the headset and measure the SNR as a function of the angluar deviation from the perfect orientation. Note that the headset rotation causes misalignment between the transmitter's and receiver's beams. Fig. 5-2 shows the SNR of the received signal versus the amount of headset rotation. The figure shows that even a minor head rotation of a few degrees can cause a major degradation in the SNR. As was shown in section 4, such reduction in SNR creates outages for the VR application. This experiment confirms the need for real-time beam tracking to realign the transmitter's and the receiver's beams as the player moves her head.

5.0.2 Beam Alignment and Tracking

In this section, we explain how MoVR aligns the transmitter's and receiver's beams, and adapts the alignment as the headset moves. Recall that MoVR operates in two different modes (as shown in Fig. 3-1). In the first mode, the AP communicates to the headset directly. This requires a beam alignment between the AP and the headset. In the second mode, the AP communicates to the headset through the mirror. This requires beam alignment between the AP and the mirror, and beam alignment between the mirror and the headset. Therefore, there are three types of beam alignment which need to be addressed in MoVR. Below, we explain each of them in more detail.

(a) Beam alignment between the AP and the mirror: To deliver the signal from the AP to the mirror, the AP needs to align its transmit beam toward the mirror and the mirror needs to align its receive beam toward the AP. Since both the AP and the mirror are static, this alignment is only done once when the mirror is installed. Though this alignment has no real-time constraints, it cannot employ past work on beam alignment [47, 41, 34, 42, 36] because all of these schemes require both nodes to transmit and/or receive signals. A MoVR mirror, however, can neither transmit nor receive; it can only reflect signals.

Thus, MoVR delegates to the AP the task of measuring the best beam angle, which the AP can then communicate to the mirror using a low-bit-rate radio, such as bluetooth. During this estimation process, the AP transmits a signal and the mirror tries to reflect this signal back to the AP itself (instead of reflecting it to the headset) allowing the AP to measure the best angle. The mirror, however, does not yet know the direction of the AP, so it has to try various directions and let the AP figure out the direction that maximizes the SNR.

Thus, our algorithm works as follows. It first sets the mirror's receive and transmit beams to the same direction, say θ_1, and sets the AP's receive and transmit beams to the same direction, say θ_2. Then it tries every possible combination of θ_1 and θ_2 while the AP is transmitting a signal and measuring the power of the reflection

(from the mirror). The θ_1 and θ_2 combination that gives the highest reflected power corresponds to the angles for the best alignment of the mirror's receive beam and the AP's transmit beam.

One problem remains. As described above, the AP needs to measure the power of the signal reflected by the mirror, while transmitting its own signal. Performing this measurement is not easy. This is due to the fact that the AP is trying to transmit and receive at the same time. As a result, the transmitted signal leaks from the AP's transmit antenna to its receive antenna. So to measure the reflected signal power, the AP first needs to separate it from the strong leakage signal it receives.

To overcome this problem, we use the fact that, if the mirror modulates the signal before it reflects it, the AP can separate the reflected signal from the leakage signal as the two signals become different. For example, if the AP transmits a sinewave at a frequency f_1, and the mirror modulates this signal by turning its amplifier on and off at a frequency f_2, then the center frequency of the reflected signal will be $f_1 + f_2$ while the leakage signal remains at f_1. Hence, the AP can simply use a filter to separate the reflected signal from the leakage signal.

(b) Beam alignment and tracking between the AP and the headset: To deliver the signal directly from the AP to the headset, the AP needs to align its transmit beam toward the headset and the headset needs to align its receive beam towards the AP. Here, we will explain how MoVR estimates the angles for best alignment of the AP's transmit beam and headset's receive beam.

We observe that VR systems have to track the location and orientation of the headset in order to update the 3D view of the player. Specifically, the HTC VR system does this using laser trackers and an IMU on the headset. Using this infrastructure, the VR system is able to calculate the headset's exact position relative to each laser tracker. By co-locating MoVR's AP with one of the VR's laser trackers, we can exploit the VR tracking system to find the exact location and orientation of the headset relative to the AP.[1] MoVR leverages this information to calculate the best

---

[1]In practice, location of the AP may be a few cm different from the location of the laser tracker. Since this is a fixed deviation, it can be calibrated by the manufacturer. Also note that even if the AP is blocked from the headset, the VR system has enough redundancy to localize the headset.
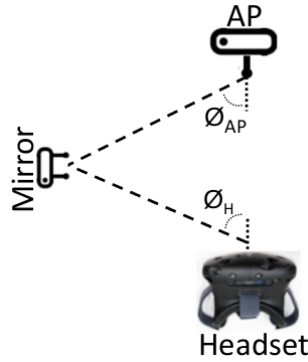
Figure 5-3: **Localizing the mirror:** MoVR finds the location of the mirror in the VR setup by intersecting the line-of-sight angle from the AP to mirror (ϕ_{AP}) and the line-of-sight angle from the headset to the mirror (ϕ_H).

alignment to the headset and track the alignment in real time.

(c) Beam alignment and tracking between the mirror and the headset: In order to align the mirror's transmit beam with the headset's receive beam, we need to know the location and orientation of the headset with respect to the mirror. Unfortunately, the VR tracking system only has this information with respect to the AP. In order to switch the reference point, we need to get the location and orientation of the mirror with respect to the AP. Getting the orientation was explained in section 5(a), but we still need to get the distance between the AP and the mirror.

Because the location of the mirror with respect to the AP is fixed during use of the VR system, one naïve solution is to ask the user to measure it during installation. However, this requires an accurate measurement, since even a small measurement error creates a significant inaccuracy in beam alignment. To avoid this, MoVR uses an automated calibration mechanism which calculates the location of the mirror with respect to the AP without any help from the user.

This calibration mechanism works by intersecting the line-of-sight angle from the AP to the mirror (ϕ_{AP}) and the line-of-sight angle from the headset to the mirror (ϕ_H), as shown in Fig. 5-3. In section 5(a), we explained how MoVR estimates ϕ_{AP}. To estimate ϕ_H, MoVR first configures the AP to transmit to the mirror. Then it tries every combination of mirror transmit beam angle and headset receive beam angle.

The receive beam angle which gives the highest SNR at the headset corresponds to ϕ_H. Finally, by intersecting the two spatial directions ϕ_H and ϕ_{AP}, we can determine the location of the mirror. Because the mirror location is fixed, this process only needs to happen once, during installation. Subsequently, MoVR can calculate the beam alignment between the mirror and the headset from VR tracking information and the mirror's known location.

Chapter 6

System Details

The last sections presented the solutions to the two challenges in using mmWave in VR systems –i.e. blockage and mobility. However, a number of system details must be addressed in order to put these solutions into practice. In particular: How do we provide connectivity at all locations within the VR space? And how do we choose between sending data via the direct link or sending by way of the mirror? This section will iron out these details and provide guidance on the system design trade-offs to maximize MoVR's coverage and performance.

How do we provide connectivity at all locations within the VR space? The system's ability to provide wireless coverage throughout the VR space is limited by the fact that the line-of-sight to the headset can be blocked by the environment and/or by the user's limbs. We addressed this problem by designing the MoVR mirror described in section 4. However, it is still possible that the headset experiences blocking along the path to both the AP and to the mirror. To address this issue, MoVR supports multiple mirrors. Each mirror adds another path to the headset, and reduces the probability of the headset being blocked exponentially. Additionally, we recommend placing multiple antennas on the headset to reduce the probability of the user's head blocking a line-of-sight path. However, any body parts that come between the headset and the AP can block all headset antennas, necessitating the use of a mirror.

How do we choose between the direct link and a link via a mirror? The AP, the mirrors, and the headset are equipped with a cheap, low-bitrate radio, e.g. Bluetooth, to exchange control information. In MoVR, the mmWave receiver on the headset continuously monitors the SNR of its received signal and, whenever it drops below a certain threshold, the headset reports it back to the AP over Bluetooth. The AP then switches to a different link. The AP picks the mirror closest to the current location of the headset. If the SNR does not go above the desired threshold, the AP switches to the next closest mirror.

Because any small period of outage impacts the quality of the data rate, the headset should act preemptively by looking at the time series of SNR and ordering a link change if there is a downward trend that is likely to result in outage. As demonstrated in section 7, the switching latency cost is sufficiently small that it does not impact the user experience, even if the AP tries more than one mirror.

Chapter 7

Evaluation

We have built a prototype of MoVR using off-the-shelf components. MoVR's mirror hardware consists of two phased array antennas (one for receive and one for transmit), connected to each other through a variable gain amplifier as shown in Fig. 4-3. The phased arrays consist of patch antenna elements, designed and fabricated on PCB. The outputs of the patch antennas are connected to Hittite HMC-933 analog phase shifters, which allows us to steer the antennas' beams. To create a variable gain amplifier, we use a Hittite HMC-C020 PA, a Quinstar QLW-2440 LNA and a Hittite HMC712LP3C voltage-variable attenuator. The mirror's current consumption is mainly dominated by it's PA, which consumes 250mA during normal operation. For controlling MoVR's mirror and measuring its amplifier's current consumption, we built a controller board using an Arduino Due micro-controller, Analog Devices DACs, and a Texas Instruments INA169 DC current sensor, as shown in Fig. 7-1.

In our experiments, we use the HTC VIVE virtual reality system. However, our design is also compatible with other high-quality VR platforms, such as Oculus Rift. [1] We equip the VIVE headset with a mmWave receiver and the VR PC with a mmWave AP working at the 24GHz ISM band [28]. The PC has an Intel i7 processor, 16GB RAM and a GeForce GTX 970 graphics card, which is required for the HTC VIVE VR setup. The transmission power is in accordance with FCC rules [35].

---

[1]Although these systems use different technologies to track the user and headset, they all provide the location and position of the headset very accurately.

Figure 7-1: **MoVR mirror's controller board:** The figures shows our custom-designed controller board for configuring the beam alignment and the amplifier gain in real time.

We evaluate MoVR in a $5m \times 5m$ office room with standard furniture, such as desks, chairs, computers and closets.[2] We perform experiments in both line-of-sight and non-line-of sight scenarios.

7.0.1 Blockage During an Actual VR Game

In section 4, we demonstrated the impact of signal blockage on the SNR and data rate of mmWave communications in a VR setup. As was shown, even blocking the signal with one's hand significantly degrades the data rate, which is problematic. In this experiment, we investigate how often the AP's line-of-sight to the headset is blocked in a realistic gaming scenario.

To do so, we ask a user to play a VR game (The Lab Solar System) while we extract the location information of the headset, access point, and two game controllers (held by the player's hands) from the VR tracking system. Using this information, we find the equation of the line between the headset and the base station. Then, as the user plays the game, we check if the locations of the user's hands (i.e. controllers' locations) lie on that line. If either hand lies on the line, we conclude that the line-of-sight path between the headset and the base station is blocked. Our results show that the line-of-sight was blocked 20 times during a 5 minute game. Fig. 7-2 plots the CDF of the durations of these 20 cases. The figure shows that the median blockage

---

[2]Our test environment is the same as what HTC VIVE recommend for operation. For safety reason, they also suggest to move the furniture outside of the game area [11].
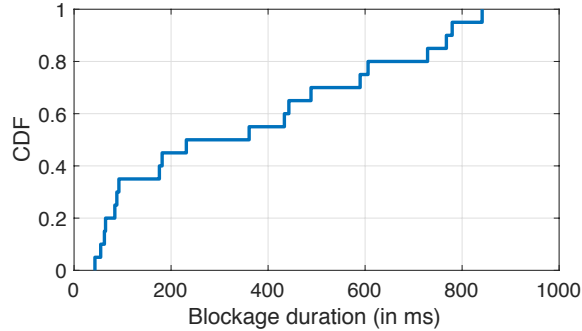
Figure 7-2: **Blockage duration:** CDF of the duration of line-of-sight blockages which happened during a 5-minute VR game. The figure shows that the median blockage durations is 245ms. This is really problematic for the VR application since the duration of it's display frame is only 10ms.

duration is 245ms. Note that the VR frame rate is only 10ms. Hence, 245ms of blockage is highly detrimental to the user's VR experience. Our results confirm that line-of-sight blockages happen very often during VR games and persist long enough to degrade the VR experience.

7.0.2 MoVR's Mirror Performance

Next, we would like to investigate how effective MoVR is in addressing the blockage problem. We place the AP on one side of the room and the mirror on an adjacent side, as shown in Fig. 3-1.[3] We place the headset at a random location and orientation. The AP transmits packets of OFDM symbols and the headset receives these packets and computes the SNR. We perform the experiment for 20 runs, changing the location and orientation of the headset for each run. We repeat each run for three scenarios:

- *No-Blockage:* In this scenario, there is a clear, direct path between the AP and the headset receiver. The AP and headset have aligned their beams along this path.

- *Blockage-without-MoVR:* In this scenario, the direct path from the AP is blocked. In the absence of a MoVR mirror, the best approach is to try to reflect the signal

---

[3]Note, MoVR does not require the user to place the mirror carefully in a specific location, since its calibration mechanism is able to automatically localize the mirror.
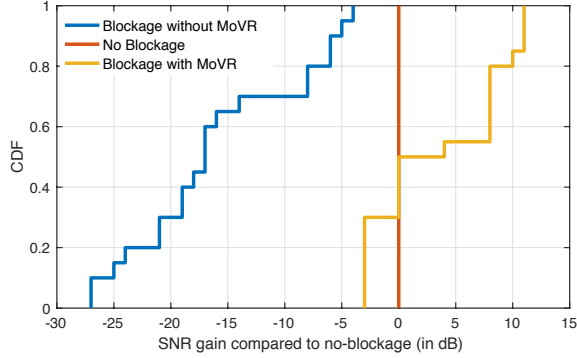
Figure 7-3: **MoVR's mirror performance:** SNR gain compared to the *No-Blockage* in all three scenarios: *No-Blockage, Blockage-without-MoVR* and *Blockage-with-MoVR.*

off of a wall or some other object in the environment. Thus, to find the best SNR possible without a mirror, we make the AP and the headset try all possible beam directions and pick the one that maximizes the SNR. The headset reports this maximum SNR.

- *Blockage-with-MoVR:* Here, we have the same blockage as in the previous scenario, but the system is allowed to use the MoVR mirror to reflect the signal as described in the earlier sections.

Fig. 7-3 compares the SNRs in all three scenarios. The figure plots the CDF of the SNR Gain relative to the SNR without blockage, defined as follows:

$$SNR\ Gain\ [dB] = SNR_{Scenario}[dB] - SNR_{No\ Blockage}[dB].$$

The figure shows that, in the absence of a MoVR mirror, a blockage drops the SNR by as much as 27dB, and the average SNR reduction is 17dB. As shown in section 4, such high reduction in SNR prevents the link from supporting the required VR data rate. Thus, simply relying on indirect reflections in the environment to address blockage is ineffective.

The figure also shows that, for most cases, the SNR delivered using MoVR's mirror is higher than the SNR delivered over the direct line-of-sight path with no blockage. This is because, in those cases, the AP's distance to the mirror is shorter than its

distance to the headset's receiver. Thus, the presence of MoVR's mirror along the path, and the fact that it amplifies the signal, counters the SNR reduction due to the longer distances to the headset. The figure further shows that, in some cases, MoVR performs 3dB worse than the no blockage scenario. This loss does not affect the data rate because, in these cases, the headset is very close to the AP, which provides a very high SNR (30dB) at the headset's receiver. This SNR is much higher than the 20dB needed for the maximum data rate. This experiment shows that MoVR's mirror enables a high data rate link between a VR headset and a PC even in the presence of blockage.

7.0.3   MoVR's Beam Alignment and Tracking Performance

As explained earlier, beam alignment is essential for mmWave links. In this section, we first investigate the accuracy of aligning the beam between the AP and the mirror. We then evaluate the accuracy of the beam alignment to the headset and the latency involved in establishing the alignment.

(a) **Accuracy of beam alignment between the AP and the mirror:** We evaluate MoVR's ability to find the best beam alignment between the AP and the mirror. We place the mirror somewhere in our testbed and estimate the angle which provides the best beam alignment between it and the AP, using the method described in setion 5.2. We repeat the experiment for 100 runs, changing the mirror location and orientation each time. We compare this to the ground truth angle, calculated from the locations of the AP and mirror. We use a Bosch GLM50 laser distance measurement tool to measure these locations to within a few millimeters.

Fig. 7-4 plots the angle estimated by MoVR versus the ground truth angle. The figure shows that MoVR estimates the angle of best beam alignment to within 2 degrees of the actual angle. Note that since the beam-width of our phased array is ∼10 degrees, such small error in estimating the angle results in a negligible loss in SNR.

(b) **Beam alignment accuracy for the whole system:** As explained in section 5, MoVR leverages the location information to track and align the transmitter's and
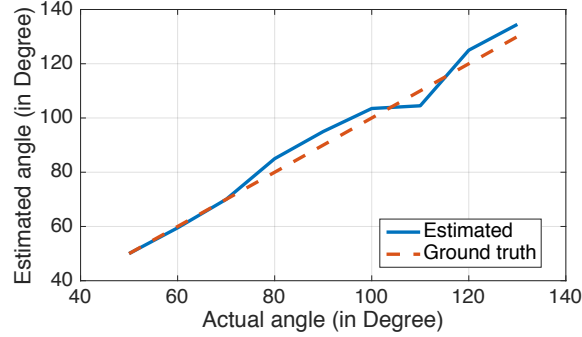
Figure 7-4: **AP to mirror beam alignment accuracy:** Angle estimated by MoVR (blue) versus the ground truth angle (red).
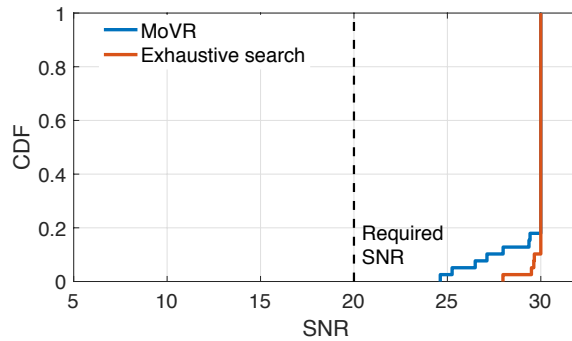
Figure 7-5: **MoVR Beam Alignment Accuracy:** The SNR of the signal at the VR receiver for two different scenarios: (1)MoVR's beam alignment algorithm and (2)exhaustive search. The figure shows that MoVR's beam alignment algorithm performs 4dB worse than exhaustive search in some cases. However, this loss does not affect the data rate, since the SNR is always much higher than the 20dB needed by the VR headset.

receiver's beams. In this experiment, we evaluate the performance of MoVR in finding the best beam alignment to the headset (either from the AP or from the mirror). We place the headset at a random location and orientation in our testbed and compute the SNR it receives in two different scenarios: (1) MoVR's beam alignment algorithm and (2) Exhaustive search, which tries all possible combinations of AP, mirror and headset beam directions and picks the set which provide the highest SNR. We repeat this experiment 40 times, changing the location and orientation of the headset each time. Fig. 7-5 plots the results of this experiment. For most cases, MoVR's algorithm performs as well as Exhaustive search in finding the best beam alignment. MoVR's beam alignment performs 4dB worse than Exhaustive search in some cases, but this

loss does not affect the data rate since the SNR is always much higher than the 20dB needed by the VR headset.[4] This result is significant since Exhaustive search requires trying all possible beam alignments, and hence introduces much latency and overhead. In contrast, given an initial calibration, MoVR obtains its alignment for free by leveraging the location information already available to the VR system.

(c) Beam alignment latency for the whole system: Next, we evaluate the capability of MoVR to perform beam alignment and tracking in real time. MoVR's beam alignment process includes multiple sources of delay, which sum to the total latency of the system. First, it takes $1ms$ for the VR tracking to update the headset position[14]. Our beam alignment algorithm, implemented in C++, uses this position to calculate new beam angles in $0.9\mu s$. Finally, MoVR's hardware (including the DACs and phase shifters) takes $1.7\mu s$ to reconfigure the beam [4, 5]. Given that our computation and the hardware reaction time are on the order of a few microseconds, the total delay is dominated by the VR location tracking delay, which is $1ms$. This delay is intrinsic to the VR system and is low enough to support the VR frame rate.

7.0.4 MoVR System Performance

Finally, we would like to evaluate the system as a whole and its ability to deliver the desired performance as the player moves around anywhere in the room. We place the headset at a random location and orientation in our VR testbed, and block the direct path between it and the AP with a hand. We then compute the SNR that the headset receives for three different scenarios: (1) No mirror, which tries all possible combinations of AP and headset beam directions and picks the one which provides the highest SNR; (2) Fixed gain mirror, where there is a mirror with a fixed amplification gain in our setup; and (3) MoVR, where we use our tracking algorithm and a mirror with our automatic gain control algorithm. We repeat this experiment 40 times, changing the location and orientation of the headset each time.

Fig. 7-6 plots the results of this experiment. The figure shows the received SNR at the headset for different room locations and for each scenario. Our results show

---

[4]The maximum SNR is limited at 30dB because of the dynamic range of the hardware.

that, in the presence of blockage, having a mirror with a fixed gain improves the SNR over relying on indirect reflections from the environment. However, there are still some locations with SNR below the 20dB needed by the VR headset. Adapting the mirror's amplifier gain improves the performance further and allows the system to achieve high SNR (24dB or higher) in all locations. This experiment confirms the need for our automatic gain control algorithm and shows that MoVR enables a high-quality untethered virtual reality, providing the required SNR for every location in a representative VR testbed.
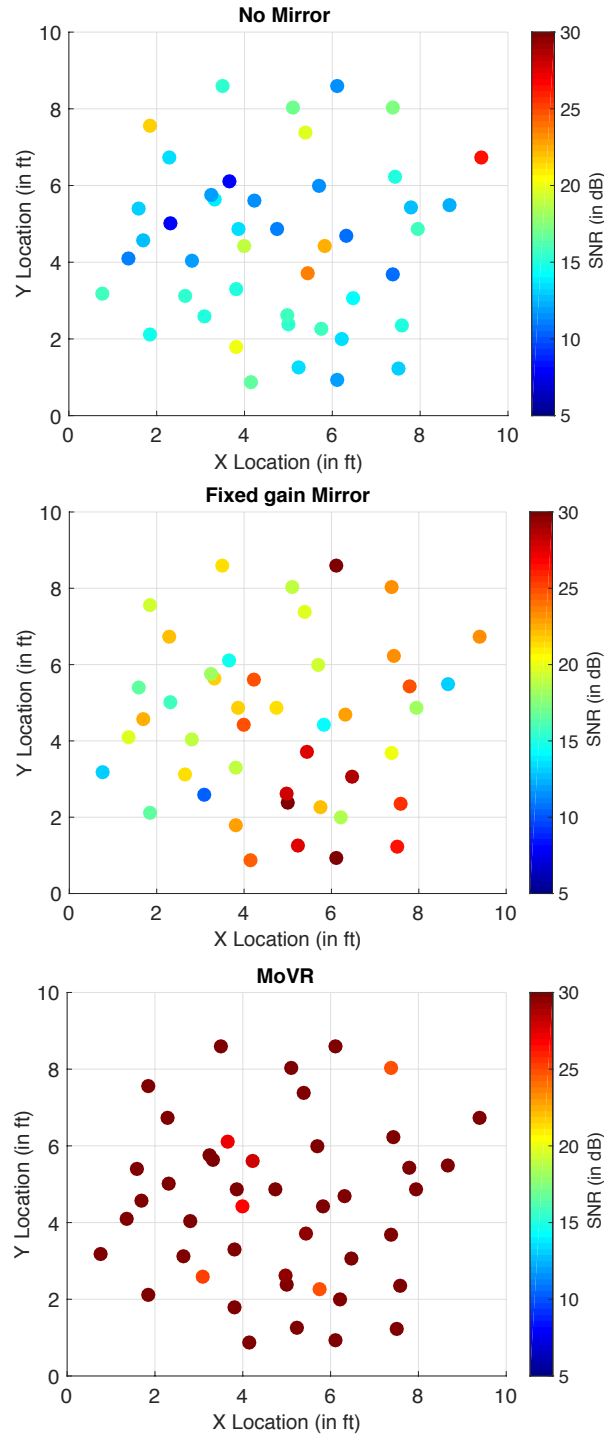
Figure 7-6: **SNR at the headset's receiver** SNR at the headset's receiver in the case of blockage for three different scenarios. (1) no mirror, (2) Fixed gain mirror, where there is a mirror with a fixed amplification gain in our setup, and (3) MoVR, where we use our beam tracking algorithm and a mirror with automatic gain control. The figure shows that during blockage the SNR is below the 20dB SNR needed by the VR headset for most locations. In contrast, MoVR enables high SNR in all locations.

Chapter 8

Concluding Remarks

This thesis presents MoVR, a system that enables a reliable and high-quality untethered VR experience via mmWave links. It provides a sustainable, high-data-rate wireless link to the VR headset even in the presence of blockage and mobility. In particular, it overcomes blockage of the mmWave link by introducing a smart and simple mmWave mirror that can reconfigure itself and adapt its angles of incidence and reflection. Further, MoVR introduces a novel algorithm that combines VR headset tracking information with RF-based localization to quickly steer the mmWave radios' beams and keep them aligned as the player moves around. Finally, it is worth mentioning that we have focused on eliminating the high-rate HDMI connection between the PC and headset. However, the current headset also uses a USB cable to deliver power. This cable can be eliminated by using a small rechargeable battery. The maximum current drawn by the mmWave radio and the HTC Vive headset is 1500mAh. Hence, a small battery (3.8x1.7x0.9in) with 5200mA capacity can run the headset for 3-4 hours [2]. An end-to-end evaluation of full system while VR data is streamed in real time and also improving the efficiency of mmWave hardware to increase the battery life are interesting avenues for future work.

Bibliography

[1] *60 GHz: Taking the VR Experience to the Next Level.* `http://www.sibeam.com/en/Blogs/2016/March/60GHZTakingtheVRExperience.aspx`.

[2] *Anker Astro 5200mAh battery.* `https://www.amazon.com/Anker-bar-Sized-Portable-High-Speed-Technology/dp/B00P7N0320`.

[3] *CNET review for Samsung Gear VR.* `http://www.cnet.com/products/samsung-gear-vr/`.

[4] *Datasheet DAC-7228.* `http://www.analog.com/media/en/technical-documentation/data-sheets/AD7228.pdf`.

[5] *Datasheet Phase Shifter HMC933.* `http://http://www.analog.com/media/en/technical-documentation/data-sheets/hmc933.pdf`.

[6] *Facebook Expects to Ship 2.6 Million Oculus Rifts by 2017.* `http://www.businessinsider.com/facebook-expects-to-ship-26-million-oculus-rifts-by-2017-2016-4`.

[7] *FCC to explore 5G services above 24 GHz.* `http://www.fiercewireless.com/tech/fcc-to-explore-5g-services-auctioned-or-unlicensed-above-24-ghz`.

[8] *HTC sold 15,000 $800 Vive virtual reality headsets in 10 minutes.* `http://venturebeat.com/2016/02/29/htc-sold-15000-800-vive-virtual-reality-headsets-in-10-minutes/`.

[9] *HTC sold 15,000 $800 Vive virtual reality headsets in 10 minutes.* `http://www.ubeeinteractive.com/sites/default/files/Understanding\%20Technology\%20Options\%20\%20for\%20Deploying\%20Wi-Fi\%20White\%20Paper.pdf`.

[10] *HTC Vive Oculus Rift Spec Comparison.* `http://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/`.

[11] *HTC VIVE Recommended Area.* `http://www.ibtimes.co.uk/htc-vive-vr-how-much-room-space-do-i-really-need-1558494`.

[12] *IEEE 802.16 Broadband Wireless Access Working Group.* `http://ieee802.org/16/maint/contrib/C80216maint-05_112r8.pdf`.

[13] *IEEE 802.16 Broadband Wireless Access Working Group.* `http://www.keysight.com/upload/cmc_upload/All/22May2014Webcast.pdf?&cc=US&lc=eng`.

[14] *Look Inside the HTC Vive's Positional Tracking System.* `http://http://www.gamasutra.com/view/news/273553/An_expert_look_inside_the_HTC_Vives_positional_tracking_system.php`.

[15] *mmWave 24GHz Transceivers.* `http://www.infineon.com/cms/en/product/rf-and-wireless-control/mm-wave-mmic/channel.html?channel=db3a304339d29c450139d8bdb700579d`.

[16] *Optoma's wireless VR headset frees you from PC cables.* `http://www.pcworld.com/article/3044542/virtual-reality/optomas-new-wireless-vr-headset-frees-you-from-pc-cables.html`.

[17] *Samsung Gear VR.* `http://www.samsung.com/us/explore/gear-vr/`.

[18] *Sulon sneak peak.* `http://sulon.com/blog/sulon-q-sneak-peek`.

[19] *Virtual Apple Airport Express.* `http://www.apple.com/airport-express/`.

[20] *Virtual Reality in Entertainment.* `http://www.vrs.org.uk/virtual-reality-applications/entertainment.html`.

[21] *Virtual Reality in Healthcare.* `http://www.vrs.org.uk/virtual-reality-healthcare/`.

[22] *Visus VR.* `http://www.visusvr.com/`.

[23] *VR for everyone.* `https://vr.google.com/`.

[24] *Wireless HDMI.* `http://www.cnet.com/news/wireless-hd-video-is-here-so-why-do-we-still-use-hdmi-cables/`.

[25] *WorldViz.* `http://aecmag.com/technology-mainmenu-35/1130-news-worldviz-brings-warehouse-scale-vr-to-unreal-and-unity-engines`.

[26] Wilocity 802.11ad Multi-Gigabit Wireless Chipset. http://wilocity.com, 2013.

[27] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Cutting the cord in virtual reality. In *HotNets*, 2016.

[28] Omid Abari, Haitham Hassanieh, Michael Rodreguiz, and Dina Katabi. Poster: A Millimeter Wave Software Defined Radio Platform with Phased Arrays. In *MOBICOM*, 2016.

[29] Dinesh Bharadia and Sachin Katti. Fastforward: Fast and constructive full duplex relays. In *SIGCOMM*, 2014.

[30] S. Boyd. *Lecture 12: Feedback control systems: static analysis.* `https://stanford.edu/~boyd/ee102/ctrl-static.pdf`.

[31] Cripps Steve C. *Advanced Techniques in RF Power Amplifier Design.* Artech House, 2002.

[32] Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Chao Zhu, Xiangyang Li, Liu Yang, and Ning Ge. Diamond: Nesting the Data Center Network with Wireless Rings in 3D Space. In *NSDI*, 2016.

[33] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory.* Courier Corporation, 2013.

[34] Mohammed E. Eltayeb, Ahmed Alkhateeb, Robert W. Heath, and Tareq Y. Al-Naffouri. Opportunistic Beam Training with Hybrid Analog/Digital Codebooks for mmWave Systems. In *GLOBESIP*, 2015.

[35] Federal Communications Commission. Title 47, code for federal regulations.

[36] Bo Gao, Zhenyu Xiao, Changming Zhang, Depeng Jin, and Lieguang Zeng. Joint SNR and Channel Estimation for 60 GHz Systems using Compressed Sensing. In *WCNC*, 2013.

[37] P. R. Gray and R. G. Meyer. Mos operational amplifier design-a tutorial overview. *IEEE Journal of Solid-State Circuits*, 1982.

[38] Muhammad K. Haider and Edward W. Knightly. Mobility resilience and overhead constrained adaptation in directional 60 GHz WLANs: protocol design and system implementation. In *MobiHoc*, 2016.

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

[40] Shuangfeng Han, ChihLin I, Zhikun Xu, and Corbett Rowell. Large-Scale Antenna Systems with Hybrid Analog and Digital Beamforming for Millimeter Wave 5G. *IEEE Communications Magazine*, January 2015.

[41] Joongheon Kim and Andreas F. Molisch. Fast Millimeter-Wave Beam Training with Receive Beamforming. *Journal of Communications and Networks*, October 2014.

[42] Dinesh Ramasamy, Subramanian Venkateswaran, and Upamanyu Madhow. Compressive tracking with 1000-element arrays: A framework for multi-gbps mm wave cellular downlinks. In *Allerton*, 2012.

[43] Theodore S Rappaport, James N Murdock, and Felix Gutierrez. State of the art in 60GHz integrated circuits and systems for wireless communications. *Proceedings of the IEEE*, 2011.

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

[45] Sanjib Sur and Xinyu Zhang. BeamScope: Scoping Environment for Robust 60 GHz Link Deployment. In *Information Theory and Application Workshop*, 2016.

[46] Sanjib Sur, Xinyu Zhang, Parameswaran Ramanathan, and Ranveer Chandra. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *NSDI*, 2016.

[47] Wenfang Yuan, Simon M. D. Armour, and Angela Doufexi. An Efficient and Low-complexity Beam Training Technique for mmWave Communication. In *PIMRC*, 2015.

[48] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H Zheng. Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *SIGCOMM*, 2012.

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