ABSTRACT

PATWARDHAN, GAURAV SHRIKRISHNA. Jamming Beamforming: A New Attack Vector in Jamming IEEE 802.11ac Networks. (Under the direction of Dr. David Thunte.)

Wireless networks by their nature and definition are vulnerable to a variety of jamming attacks. With the recent release of the newest standard of the 802.11 family, the 802.11ac, there are new mechanisms like beamforming being introduced and standardized, which are susceptible to different types of jamming attacks. Our work summarizes those attacks and provides a proof-of-concept based implementation for one such attack. As far as we know, there has not been any previous work done using the beamforming mechanism as a jamming attack vector. We have used a 802.11ac Access Point (AP) along with a commercially available 802.11ac Netgear (A6200) client, both capable of beamforming. We show that jamming the beamforming mechanism is very effective in reducing the aggregate throughput of the targeted Wireless LAN system (WLAN). For the jammer, we are using a commercially available Universal Software Radio Peripheral (USRP2) Software Defined Radio (SDR) made by Ettus Research LLC. This SDR provides the flexibility for implementation to jam targeted packets in the beamforming process. The 802.11ac capable AP on a reference board, provided by Broadcom Corporation, is needed to access the firmware Application Programming Interface (API) to control the beamforming mechanism and to measure the effect of jamming. Our implementation is limited in scope since the standard was just published in the January of 2014. The work done in this thesis is largely proof-of-concept of a new beamforming based attack vector, since little 802.11ac beamforming capable hardware is currently available in the market.

© Copyright 2014 by Gaurav Shrikrishna Patwardhan

All Rights Reserved

Jamming Beamforming: A New Attack Vector in Jamming IEEE 802.11ac Networks

by
Gaurav Shrikrishna Patwardhan

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Networking

Raleigh, North Carolina

2014

APPROVED BY:

Dr. Mihail Sichitiu

Dr. Khaled Harfoush

Dr. David Thuent
Chair of Advisory Committee

DEDICATION

To my parents Shirish and Milan,
To my late grandfather Dr. R. R. Ambardekar,
to my sister Amruta,
and my fiancée Shireen.

BIOGRAPHY

Gaurav Patwardhan was born in Pune, India in 1988. He graduated with Bachelor of Engineering (BE) degree in Electronics from Pune University in 2011. From 2011 to 2012 he worked at AirTight Networks as a Member of Technical Staff in the Quality Assurance group. He then joined North Carolina State University (NCSU), USA to pursue a Master's degree in Computer Science with specialization in Computer Networks. He worked as an Engineering Intern with Broadcom Corporation from June 2013 to May 2014 in the WLAN Quality Assurance team.

ACKNOWLEDGEMENTS

I would like to acknowledge the guidance and inspiration of my adviser, Dr. David Thuenté whose years of experience in this field proved to be invaluable. I would like to thank Dr. Mihail Sichitiu for teaching me the wireless domain in an intuitive and an informative manner without whom I would not have developed a deep appreciation for the design of wireless systems. I am grateful to both, Dr. Mihail Sichitiu and Dr. Khaled Harfoush for agreeing to be on the advisory committee and providing valuable suggestions for finishing this thesis.

I would also like to acknowledge the support of my friends and family without whom this thesis would not have been completed.

Last but not least, I appreciate the courtesy shown by Broadcom Corporation by lending me the 802.11ac equipment which formed the basis of my thesis. I would also like to thank my team lead at Broadcom Corporation, Mr. Balakrishnan Ramachandran whose insight in the 802.11 protocols proved invaluable.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
Chapter 1 Introduction	1
1.1 Motivation	2
1.2 Organization of the thesis	2
Chapter 2 Literature Survey	3
2.1 Types of Jammers and Jamming Attacks	3
2.1.1 Jammers without the knowledge of the MAC layer protocol	3
2.1.2 Jammers having the knowledge of the MAC layer protocol	4
2.2 Attacks on 802.11 networks	4
2.2.1 Identity Attacks	5
2.2.2 Media Access Attacks	5
2.2.3 Intelligent Jamming	6
Chapter 3 Introduction to 802.11ac	7
3.1 The PHY Layer	8
3.1.1 Extended MIMO	8
3.1.2 Radio Channels	9
3.1.3 Modulation and Coding	10
3.1.4 PHY Layer Framing	12
3.2 The MAC Layer	14
3.2.1 MAC Layer Framing	14
3.2.2 Management Frames	15
3.2.3 Control Frames	15
3.2.4 Medium Access Mechanisms	15
3.2.5 Security	18
3.3 Beamforming in 802.11ac	18
3.3.1 Null Data Packet Explicit Beamforming in 802.11ac	20
3.3.2 Frames used for Beamforming	21
3.3.3 Single User Beamforming	27
3.3.4 Multi User Beamforming	28
3.3.5 Computation of Feedback Matrix	29
Chapter 4 Software Defined Radio, USRP2 and GNUradio	34
Chapter 5 Jamming Attacks using Beamforming	36

Chapter 6	Setup and Implementation	38
6.1	Setup	38
6.2	USRP2 software setup	41
6.3	Access Point setup	41
6.4	Wireshark sniffer setup	42
6.5	Client setup	42
6.6	Implementation	42
6.7	Expected results	46
Chapter 7	Readings and Observation	47
Chapter 8	Conclusion and Future Work	57
BIBLIOGRAPHY		59
APPENDIX		61
Appendix A	Reference Tables	62
A.1	Compressed Beamforming Report	62
A.2	MU Exclusive Beamforming Report	66

LIST OF TABLES

Table 3.1	Difference between 802.11n and 802.11ac [Gas13]	8
Table 3.2	Channel width with subcarrier details [Gas13]	9
Table 3.3	Map of the available channels [Gas13]	10
Table 3.4	MCS rates for 802.11ac [EPne]	11
Table 3.5	Invalid MCS rates [EPne]	12
Table 7.1	Comparison of transmitted packets with and without beamforming	47
Table 7.2	Comparison of number of packets sent at various MCS rates with and without jamming	48
Table 7.3	UDP frames beamformed (not jammed v/s jammed)	49

LIST OF FIGURES

Figure 3.1	VHT Format Preamble [Gas13] [EPne]	13
Figure 3.2	A-MPDU frame format [Gas13]	14
Figure 3.3	Primary and Secondary channels [Gas13]	16
Figure 3.4	Dynamic bandwidth using RTS/CTS [Gas13]	17
Figure 3.5	Advantage of beamforming [EPne]	19
Figure 3.6	NDPA frame structure [Gas13]	22
Figure 3.7	NDP frame structure [Gas13]	23
Figure 3.8	Beamforming report poll frame structure [Gas13]	23
Figure 3.9	SU Compressed beamforming action frame [Gas13]	24
Figure 3.10	MU Compressed beamforming action frame [Gas13]	25
Figure 3.11	SU beamforming sounding sequence [Gas13].	28
Figure 3.12	MU beamforming sounding sequence [Gas13].	28
Figure 3.13	MIMO system with transmit beamforming [Gas13].	29
Figure 3.14	MU-MIMO data transmission [Gas13].	32
Figure 6.1	The experimental setup	39
Figure 6.2	Setup schematic	40
Figure 6.3	802.11 a/g OFDM receiver for GNURadio [Blo13]	43
Figure 6.4	Full flow graph of jammer implementation in GNURadio	44
Figure 6.5	Jammer part of the entire implementation	45
Figure 7.1	Graphs for ϕ_{21} angle	51
Figure 7.2	Graphs for ψ_{21} angle	52
Figure 7.3	Graphs for the absolute difference between averages for ϕ_{11} , ψ_{31} , ϕ_{22} and ψ_{32}	53
Figure 7.4	Difference between the mean of ϕ_{21} before and during jamming when packets are split up	54
Figure 7.5	Difference between the standard deviation of ϕ_{21} before and during jamming when packets are split up	54
Figure 7.6	Beamformed data transmission for a MU-MIMO system [Gas13]	55
Figure A.1	Compressed Beamforming Report Information [2].	63
Figure A.2	Order of angles in the Compressed Beamforming Feedback Matrix subfield[2].	64
Figure A.3	Subcarriers for which a Compressed Beamforming Feedback Matrix subfield is sent back [2].	65
Figure A.4	Average SNR of Space-Time Stream subfield [2].	65
Figure A.5	MU Exclusive Beamforming Report information[2].	67

Chapter 1

Introduction

Wireless networks are seen everywhere because of their ease of deployment and their usefulness in many different scenarios. Since they allow the user to roam and also provide very high data rates, they are gaining more and more importance every day. The first 802.11 standard was introduced in 1999 after which newer standards in the 802.11 family have been created and ratified at an alarming rate. The latest member of this family is the 802.11ac standard [2], which was published in January, 2014. With this standard comes the first ever implementation of gigabit wireless speeds, known as Very High Throughput (VHT). Along with other features, 802.11ac implements a standardized method for beamforming. This beamforming is susceptible to a phenomenon called ‘jamming’. Jamming results from the basic premise of a shared medium in wireless networks. It can occur due to devices such as microwave ovens, which share the same 2.4 Ghz Industrial, Scientific and Medical (ISM) radio band with other wireless devices. Jamming can also be implemented manually with malicious intentions by entities are known as jammers [BS03]. This 802.11 beamforming mechanism [1][2] is based on the sounding signal sent by the beamformer to the beamformee who, after sensing the channel, responds back to the beamformer with the nature of the channel it perceives [1][2]. The beamformer then uses this information to increase the aggregate Signal-to-Noise Ratio (SNR) at the beamformee’s antennas, and can use a higher modulation scheme, thus increasing throughput. The sounding signal sent by the beamformer is susceptible to jamming, and it is this phenomenon that we are going to explore in the following chapters. This beamforming mechanism is standardized, and therefore the work done here is platform

agnostic as we are trying to attack the weakness in the standard itself. As far as we know, this is the first implementation of its kind where beamforming is considered as an attack vector for jamming, and most certainly, the first such attack on the 802.11ac protocol.

1.1 Motivation

The 802.11 standards have time and again been criticized for their lack of comprehensive security solution that protects all the entities in the WLAN and also secures the medium. In August 2001 [Flu01], the Wired Equivalent Privacy (WEP) security mechanism was broken and has been deprecated from 2004 onwards. Even though stronger security mechanisms like Wi-Fi Protected Access (WPA) and WPA2 were created for sending data over the wireless network securely, the control and management overhead required for the proper functioning of the WLAN still faced the threat of being used as an attack vector in jamming. With that in mind, the 802.11w amendment to the 802.11 standard was published in September 2009 [11], which protected the management frames. Even then, with the published standard 802.11ac [2], the new feature of beamforming, which uses management frames, was left out of this protection mechanism. We show, in our work, that not only jamming the beamforming mechanism is a hindrance for a WLAN, but it also reduces the throughput in the network by a substantial amount. Since the future ‘direction’ of the WLAN architecture is towards MU-MIMO (Multi User Multiple Input Multiple Output), we will also show that this jamming attack will also affect those networks, and that the effect there will be compounded.

1.2 Organization of the thesis

In Chapter 2 some existing studies on WLAN attacks by jamming are considered. Then, in Chapter 3, a brief introduction to the new 802.11ac standard is given, that is followed by Chapter 4, which gives a primer on Software Defined Radios (SDRs) and Universal Software Radio Peripheral (USRP2). After that we describe different attack vectors that can be implemented using beamforming mechanism for jamming in Chapter 5. In Chapter 6 we describe how the equipment is setup and the readings taken, after which we analyze the readings in Chapter 7, and then we draw conclusions from them [Chapter 8].

Chapter 2

Literature Survey

In this chapter we briefly survey the literature about different types of jammers and jamming attacks. The later part of this chapter is dedicated specifically to the jamming attacks found in 802.11 networks.

2.1 Types of Jammers and Jamming Attacks

A jammer is an entity which shares the medium with legitimate users and interferes with their access to the network resources. A jammer may or may not be a malicious entity. For example, a microwave oven which works in the 2.4 Ghz spectrum creates noise in that spectrum, effectively lowering the throughput of any WLANs in the vicinity, and for all intents and purposes, can be called a jammer. For all chapters henceforth, we will define a jammer as a malicious entity with an intent to disrupt the normal working of the stations in a WLAN system. Jammers could be broadly classified into two categories:

2.1.1 Jammers without the knowledge of the MAC layer protocol

Jammers in this category jam the medium (involves jamming of the radio band) without having any knowledge of the Media Access Control (MAC) layer protocol which runs on top of the Physical (PHY) layer. They are further subdivided into the following categories [Pel11]:

1. Constant Jammer - This type of jammer continuously jams the spectrum to avoid any

kind of communication between nodes. Although effective, it can be detected and with different techniques, it can also be eliminated.

2. Periodic Jammer - This type of jammer, although less detectible, can be nevertheless effective, since it consumes much less energy.
3. Random Jammer - Although, over longer periods of time, the energy efficiency of this jammer would be same as that of a periodic jammer, this type of jammer is the least detectible since it jams without any recognizable pattern.

2.1.2 Jammers having the knowledge of the MAC layer protocol

Jammers in this category are aware of the protocols which the network implements, and they exploit the vulnerabilities in the inherent working mechanisms in the protocol. These jammers are also called as reactive [Xu05] or intelligent [TA06] jammers. In this category, there are some specific subtypes:

1. Spoofing - This type involves creating and injecting MAC layer frames which appear as though they originate from a legitimate source. Most of the times this is used to either reserve the medium (denial of medium access [Pel11]) or for fake management overhead, which is normally used for managing different stations.
2. Corrupting - This type involves corrupting the bits in frames sent on the medium by legitimate users and can prove devastating if the underlying physical layer is not robust enough.

2.2 Attacks on 802.11 networks

The original 802.11 standard was published in 1999, and since then many amendments have been made, the most recent one being the 802.11ac standard. The standard, as it stands today, makes the 802.11ac protocol and its communication vulnerable to many types of attacks. These attacks can be broadly classified as follows:

2.2.1 Identity Attacks

This category of attacks uses Management type frames defined in the 802.11 standard [1]. Some notable attacks using such frames are the deauthentication attack, disassociation attack and the attack on the power saving mechanism [BS03]. By using a deauthentication or a disassociation attack, an attacker can disconnect a legitimate station from an AP, thus denying it access to that basic service set. The power save mechanism was introduced in 802.11 so that wireless devices, which normally run on battery power, can conserve their energy. So, to conserve energy, a wireless station tells an Access Point (AP) of its intentions to go to 'sleep' so that the AP can buffer the inbound frames to the client. On awakening, the client polls the AP to retrieve the stored frames. By spoofing the polling message, an attacker can cause the AP to discard the buffered packets when the client is still asleep [BS03].

2.2.2 Media Access Attacks

This category of attacks focus on denying medium access to legitimate wireless clients by exploiting the channel access mechanisms. The 802.11 standard and 802.11e standard [12] define inter-frame spacings for prioritizing channel access for the wireless stations. Short Interval Frame Spacing (SIFS) and Distributed Coordination Function Interframe Space (DIFS) are the two most important ones. The SIFS is used in between frames that are a part of a sequential frame exchange. The DIFS is used by the stations to initiate a new frame exchange. The stations begin the exchange by generating a random backoff for accessing the channel, given according to DCF [1]. By sending a short signal after every SIFS or DIFS period, an attacker may completely disable the channel. Although effective, this attack requires the attacker to expend a lot of energy [BS03]. A jammer might also gain access to the channel by selecting smaller average backoff values in the DCF mechanism, and/or by using a different strategy for selecting a backoff window after a collision [KV03]. The hidden terminal scenario poses a threat to a WLAN without the existence of an attacker. Request to Send (RTS) and Clear To Send (CTS) control frames have been designed to prevent such a scenario. The RTS reserves the channel at the transmitter's end and the CTS reserves it at the receiver's end. To set the Network Allocation Vector (NAV) [1], a duration field is used in these frames. The NAV allows virtual carrier sensing and hence keeps the channel clear for the entire frame

exchange. By spoofing the RTS and the CTS frames with large duration values to ‘reserve’ the channel (maximum is 32ms [BS03]), a jammer can effectively create a Denial of Service (DOS) attack on the medium.

2.2.3 Intelligent Jamming

These attacks mostly involve sniffing the wireless traffic and then reacting to it by corrupting specific frames to make an entire set of transaction of frames void. CTS, Acknowledgement (ACK) and Data corruption attacks fall into this category [TA06]. The attack typically involves corrupting some bits in a frame that is a part of an exchange of sequential frames. Although these require an intelligent jammer, their effect can be devastating.

We have seen some of the types of jammers and the attacks they can do on a 802.11 WLAN. With every new standard in the 802.11 family, new mechanisms are introduced for increasing throughput, etc. Beamforming is one such mechanism that was introduced in 802.11n, but due to the lack of standardization of 802.11n beamforming in industry, it was formally standardized in 802.11ac. This mechanism is subject to jamming attacks and we now proceed to demonstrate how effective this jamming attack is.

Chapter 3

Introduction to 802.11ac

When 802.11 was first standardized in 1997, the wireless networks ran on 1 Mbps speed, with a really fast network running at a speed of 2 Mbps. Then 802.11b standard was released, which implemented Direct Sequence Spread Spectrum (DSSS) and supported 11 Mbps. This was followed by substituting the single carrier technology with a multiple carrier one, known as Orthogonal Frequency Division Multiplexing (OFDM), used in the 802.11a and 802.11g standards. From there the standard moved on to implementing Multiple Input Multiple Output (MIMO) in 802.11n. Finally, the 802.11ac standard was created which pushed the wireless throughput to gigabit speeds (VHT). This chapter gives an overview of the 802.11ac technology, and then explains in depth on the beamforming mechanism. Conceptually, 802.11ac is an evolved standard from 802.11n and is not a revolutionary change in the standard like 802.11n was to 802.11a and 802.11 b/g. Most of the techniques, which are in 802.11n, are just ‘polished’ in 802.11ac with two exceptions. The Multi-User Multiple Input Multiple Output (MU-MIMO) technique from the 802.11ac standard is entirely new to the 802.11 family. Also, 802.11ac supports ONLY the 5 Ghz frequency band since it allows 80 Mhz and 160 Mhz channels, whereas 2.4 Ghz does not support those channel sizes. Some enhancements/differences to 802.11n made by 802.11ac are shown in Table 3.1. (The background for the material in this entire chapter is [Gas13],[EPne],[2]).

Table 3.1 Difference between 802.11n and 802.11ac [Gas13]

802.11n	802.11ac
Supports 20 and 40 MHz channels	Adds 80 and 160 MHz channels
Supports 2.4 GHz and 5 GHz frequency bands	Supports 5 GHz only
Supports BPSK, QPSK, 16-QAM, and 64-QAM	Adds 256-QAM
Supports many types of explicit beamforming	Supports only null data packet (NDP) explicit beamforming
Supports up to four spatial streams	Supports up to eight spatial streams (AP); client devices up to four spatial streams
Supports single-user transmission only	Adds multi-user transmission
Includes significant MAC enhancements (A-MSDU, A-MPDU)	Supports similar MAC enhancements, with extensions to accommodate high data rates

3.1 The PHY Layer

To increase raw speed of the PHY layer in 802.11ac three things have been done:

1. Increased the number of MIMO streams.
2. Used wider channels.
3. Created better modulation technique (to get more bits/symbol).

Lets see how these have been designed in the 802.11ac standard.

3.1.1 Extended MIMO

One of the major techniques used in 802.11ac to increase the throughput as compared to 802.11n is doubling from four to eight the number of spatial streams. This protocol feature directly doubles the throughput over an equivalent 802.11n system.

Table 3.2 Channel width with subcarrier details [Gas13]

PHY standard	Subcarrier range	Pilot subcarriers	Subcarriers (total/ data)	Capacity relative to 802.11a/g	Capacity relative to 20 MHz 802.11ac
802.11a/g	-26 to -1, +1 to +26	$\pm 7, \pm 21$	52 total, 48 usable (8% pilots)	x1.0	n/a
802.11n/ 802.11ac, 20 MHz	-28 to -1, +1 to +28	$\pm 7, \pm 21$	56 total, 52 usable (7% pilots)	x1.1	x1.0
802.11n/ 802.11ac, 40 MHz	-58 to -2, +2 to +58	$\pm 11, \pm 25, \pm 53$	114 total, 108 usable (5% pilots)	x2.3	x2.1
802.11ac, 80 MHz	-122 to -2, +2 to +122	$\pm 11, \pm 39, \pm 75,$ ± 103	242 total, 234 usable (3% pilots)	x4.9	x4.5
802.11ac, 160 MHz	-250 to -130, -126 to -6, +6 to +126, +130 to +250	$\pm 25, \pm 53, \pm 89,$ $\pm 117, \pm 139,$ $\pm 167, \pm 203,$ ± 231	484 total, 468 usable (3% pilots)	x9.75	x9.0

3.1.2 Radio Channels

The 802.11ac brings two new channel sizes, 80 Mhz and 160 Mhz and uses OFDM based transmission with each subcarrier having a bandwidth of 312.5 kHz. The 80 Mhz channels are contiguous blocks of spectrum, whereas, due to the difficulty of finding a 160 Mhz contiguous channel, the 160 Mhz block can be split if required into two 80 Mhz non-contiguous blocks of spectrum. Table 3.2 gives the list of subcarriers in all the 802.11 standards. In OFDM, not all subcarriers are used for carrying data. Some of the subcarriers are used for equalizing the gain as well as determining the phase shift at the receiver's side. These are known as pilot subcarriers. As we can see, the percentage of pilot subcarriers decreases as we go from 802.11a to 802.11ac at 160 Mhz. This results in greater efficiency as the overhead of the pilot subcarriers is reduced. Allocation and definition of channels is more of a regulatory

Table 3.3 Map of the available channels [Gas13]

concern than actual device capability. Since there is no 160Mhz or 80 Mhz channel practically available in 2.4 Ghz band, 802.11ac is defined only for 5 Ghz frequency band. Table 3.3 shows the available channels. The dark area shows the frequency already approved by the Federal Communications Commission (FCC) whereas the gray area shows the proposed channels, which are pending FCC approval. As we can see, without the gray channels, there are not many 160 Mhz channels of which 802.11ac can take advantage.

3.1.3 Modulation and Coding

In 802.11ac, modulation and coding are coupled together into a single number, the Modulation and Coding Scheme (MCS) index. 802.11ac has made some changes in the listing of MCS rates as well as how they are selected for a particular transmission. The number of MCS

Table 3.4 MCS rates for 802.11ac [EPne]

MCS index value	Modulation	Code rate (R)
0	BPSK	1/2
1	QPSK	1/2
2	QPSK	3/4
3	16-QAM	1/2
4	16-QAM	3/4
5	64-QAM	2/3
6	64-QAM	3/4
7	64-QAM	5/6
8	256-QAM	3/4
9	256-QAM	5/6

rates has been brought down from 70 as given in 802.11n [1] to 10 [2] and are shown in Table 3.4. Unlike 802.11n [1], the MCS rates have been decoupled from the channel bandwidth and spatial streams, so, to determine link speed, the known MCS rate must be combined with the channel width and spatial streams to give the overall data rate. Unequal modulation on each spatial stream was supported by 802.11n to support transmit beamforming, but is now discarded in 802.11ac. It has been made mandatory by the standard to implement from MCS rate index 0 up to and including MCS rate index 7 with most of the vendors going all the way to MCS rate index 9. The 802.11ac standard has several MCS values that are not used, because these combinations of MCS rates and channel widths do not cleanly fit the boundaries of encoding and interleaving process to create a frame. They are given in Table 3.5. Previous standards allowed modulation up to 64-QAM (Quadrature Amplitude Modulation), which allowed each transmission symbol to take one of 64 values. 802.11ac introduces 256-QAM that increases the PHY layer link speed by 33% over its nearest equivalent rate in 802.11n. But, to achieve this speed there should be much higher Signal-to-Noise Ratio (SNR) (about 5 dB more) than what is required for 64-QAM. This gap is bridged using a number of different techniques. One of them is the introduction of a new error correcting code mechanism called

Table 3.5 Invalid MCS rates [EPne]

MCS Rate	Channel Bandwidth (Mhz)		
	20 Mhz	80 Mhz	160 Mhz
MCS 6	N/a	3 and 7 Spatial Streams	N/a
MCS 9	1,2,4,5,7, and 8 Spatial Streams	6 Spatial Streams	3 Spatial Streams

Low-Density Parity Check (LDPC) that can provide a gain of up to 1-2 dB. Also some RF front-end techniques may be used to increase the SNR at the receiver's antennas.

3.1.4 PHY Layer Framing

The 802.11ac PHY is designed in such a way that it is compatible with previous 802.11 PHYs. When a frame is transmitted, 802.11a and 802.11n devices must be able to see and defer their transmission for the time required by the frame sent on the medium. To meet this requirement, the format of the VHT physical layer frame is similar to the mixed-mode format used in 802.11n (Figure 3.1). To enable MU-MIMO transmissions, the preamble needs to be designed to describe the number of spatial streams and enable multiple receivers to set up to receive their frames. To meet this second requirement, a new physical layer header was required because the 802.11n High Throughput Signal (HT-SIG) header field was not extensible to new channel widths or large numbers of spatial streams. Compared to 802.11n [1], the physical layer for 802.11ac [2] is much simpler because there is only one format. Like in 802.11n [1], the first part of the VHT Format Preamble consists of legacy training fields. These include the Legacy Short Training Field (L-STF), Legacy Long Training Field (L-LTF), and the Legacy Signal field (L-SIG). These fields are identical to the ones used in 802.11a and are used to identify the start of the frame, to synchronize timers and to select antennas. For every training field, the subcarriers used are extended by 802.11ac to include 80 Mhz and 160 Mhz as well. The L-SIG field has the same structure as used in 802.11n but it is used in a different way. In 802.11n, the length field in L-SIG is set such that legacy 802.11a/g devices

Figure 3.1 VHT Format Preamble [Gas13] [EPne]

properly defer their transmissions. Since the VHT Signal A (VHT-SIG-A) field does not contain any duration information, the L-SIG in a VHT frame is used not to convey the length, but the number of data symbols in the VHT frame. The value in this length field is generated in a way so that not only 802.11a/n devices defer correctly, but 802.11ac devices can also calculate the length of the PHY frame accurately. The VHT portion of the preamble contains the VHT-SIG-A, VHT-STF, VHT-LTF and VHT-SIG-B. VHT-SIG-A is transmitted at a legacy rate while the remaining portion is transmitted using VHT modulation. The VHT-STF, VHT-LTF and VHT-SIG-B fields are only understood by 802.11ac devices. The antenna weights and the automatic gain controller setting is set using the VHT-STF. The VHT-LTFs are used for finer frequency synchronization and are also used to equalize the VHT-SIG-B and the VHT Data fields. The VHT-SIG-A and the VHT-SIG-B fields are used to give the frame attributes such as modulation and coding, channel width, Single User (SU) or Multi-User (MU) frame, whether Space Time Block Coding (STBC) encoding is applicable or not, number of space-time streams, whether the packet is beamformed or not, etc. It is interesting to note that VHT-SIG-B always takes up one symbol time on the medium irrespective of the channel bandwidth. This is done by repeating the field to create enough length to occupy one symbol time on the channel [Gas13]. After the physical layer header, the Service field along with the Data field, the PHY pad and the Tail are transmitted using VHT modulation that is specified in the PHY layer header. The Service field is used to initialize the data scrambler to avoid long runs of identical bits and it also contains the Cyclic Redundancy Check (CRC) of the VHT-SIG-B field to detect errors. Following the Service field, the Data field is sent which is

variable in length and is padded at the end with the PHY pad field. The Tail bits at the end are used to reset the convolutional encoder, which enables it to receive the next incoming frame.

3.2 The MAC Layer

The MAC layer enhancements in 802.11ac are mostly derived from the 802.11n and their function is to support the new PHY layer features. The MAC layer, which also caters to channel access methods, has undergone a large change to accommodate sharing of radio resources on channels of different sizes.

3.2.1 MAC Layer Framing

Frame aggregation was introduced in 802.11n and it helped increase the throughput by reducing the channel access control overhead. 802.11n described multiple types of aggregation like Aggregated MAC Protocol Data Unit (A-MPDU) and Aggregated MAC Service Data Unit (A-MSDU), which were optional. To simplify things, 802.11ac describes only a single type of aggregation, A-MPDU, which is used on all frames, i.e., it is not optional. Figure 3.2 shows the format of the A-MPDU. The maximum length of A-MPDU is controlled by value of the

Figure 3.2 A-MPDU frame format [Gas13]

field 'Maximum Length of A-MPDU exponent' which describes the maximum length of the A-MPDU by the formula $(2^{13+exponent} - 1)$ bytes. The value of the exponent ranges from 0 to 7 that allows a maximum A-MPDU length to be from 8 KB to 1 MB. The maximum Physical Layer Service Data Unit (PSDU) size possible is 4,692,480 bytes [Gas13].

3.2.2 Management Frames

Management Frames are used for managing the stations connecting to the wireless network. In 802.11ac, the Probe Request and Probe Response management frames have been changed to include the VHT capabilities information element. This element describes the protocol elements and speeds the transmitter is capable of using. Some of them are Maximum MPDU Length, Supported Channel Width set, Short Guard Interval (SGI), etc.

3.2.3 Control Frames

Two new types of control frames have been introduced in 802.11ac, the Null Data Packet Announcement (NDPA) and the Beamforming Report Poll frame. Both of these are used in the control overhead for the beamforming mechanism.

3.2.4 Medium Access Mechanisms

With newly introduced channel bandwidths in 802.11ac, new rules which determine whether the channel is clear or not are also established. Also to make efficient use of the spectrum, intended bandwidth requirement is included in the RTS/CTS frames.

3.2.4.1 Clear Channel Assessment (CCA)

In 802.11ac, the Basic Service Set Identifier (BSSID) can switch channel bandwidth, as required per frame. This has been implemented because large number of devices having different capabilities might connect to a single Access Point (AP). So, whenever a larger channel is available, the AP uses it, thus allowing for higher data rates. The terminology for primary and secondary channel is shown in Figure 3.3. A station in a network configured as shown in Figure 3.3 will transmit a frame on 160 Mhz if the entire spectrum from channel

Figure 3.3 Primary and Secondary channels [Gas13]

36 to channel 64 is free. If there is another transmission going on, in say, channel 44, then it will only transmit on the 80 MHz primary channel (from channel 52 to channel 64) and so on. 802.11ac can use spectrum more efficiently than 802.11n because detection of networks on non-primary channels is much better in 802.11ac hardware. 802.11n's CCA capabilities on the secondary channel required that the two overlapping networks have the same primary channel. Due to the enhanced CCA in 802.11ac hardware, it is not required for two overlapping networks to have the same primary channel. This leads to higher throughput overall, since many more transmissions can be done in parallel. Reduced Inter Frame Spacing (RIFS) which was introduced in 802.11n has been discontinued in 802.11ac since, after A-MPDU was made mandatory, the improvement in throughput using RIFS was negligible. 802.11n introduced a greenfield mode which saved a few microseconds in airtime by reducing the preamble such that only 802.11n devices could understand, i.e. there was no legacy preamble which would have enabled legacy devices to defer their transmissions. Although it was slightly more efficient, 802.11n only networks were rare, and hence for protection, a device used to send a CTS-to-self before transmitting in greenfield mode. The efficiency gains from the greenfield mode were nulled by the CTS-to-self, and hence this mode was dropped from the 802.11ac standard.

3.2.4.2 Protection and Coexistence

Protection and coexistence is not really a problem since all the channel access is deferred using the VHT-PHY, which we saw before. Since 802.11a, 802.11n and 802.11ac all use OFDM

in the PHY layer, the Legacy preamble in the 802.11ac frames makes sure that all the legacy devices can sense that the channel is busy.

3.2.4.3 Dynamic Bandwidth Feature

802.11ac standard added a 'bandwidth signalling' feature to the RTS and CTS to clear out multiple channels simultaneously. This is done by setting the 'Group' bit in the transmitter's address to '1'. The transmitter's address is then known as 'Bandwidth Signaling Transmitter Address'. The bandwidth requirement is sent through the scrambling sequence which goes in the Service field immediately after the PLCP header. This dynamic bandwidth operation is shown in Figure 3.4. In Figure 3.4, a transmitter transmits four 20 Mhz RTS frames on four

Figure 3.4 Dynamic bandwidth using RTS/CTS [Gas13]

adjoining channels to reserve a total of 80 Mhz for transmission. When the receiver does a CCA, it realizes that only the first two channels are free (a total of 40 Mhz) and responds with two CTS frames on two adjoining 20 Mhz channels reflecting its CCA result. Then instead of sending out data on 80 Mhz, the data is sent out on a channel of 40 Mhz.

3.2.5 Security

With the 802.11i amendment in 2004, the Robust Security Network (RSN) concepts came into being. The Counter Mode with CBC-MAC Protocol (CCMP), which the RSN defines, has proven to be a durable cryptographic system. CCMP uses AES (Advanced Encryption Standard) in the counter mode and then applies a cipher-block chained message authentication code [1]. CCMP uses a block-chaining mode for authenticating each block which cannot be parallelized, so there is a performance limit placed on it. Due to concerns about high latency the Galois Counter Mode Protocol (GCMP) was specified [Gas13]. At a high level, GCMP is functionally similar to CCMP. Rather than using block chaining to authenticate each data block, GCM uses a Galois field multiplication. In contrast to block chains that require each block to be processed before moving on to the next one, Galois field multiplications can be run in parallel. In addition, Galois multiplications are less computationally intensive than the cipher block encryption algorithms required by a CBC-MAC. This will be huge help when 802.11ac with MU-MIMO arrives in the market (wave 2 rumored to be out in the end of 2014), since the CCMP will affect the latency of the network at gigabit speeds. This feature is not widely used in 802.11 wireless LANs and may be used in 802.11ac, but it is not mandatory [Gas13].

3.3 Beamforming in 802.11ac

Traditionally Access Points (APs) have omnidirectional antennas that allow the propagation of electromagnetic radiation in all directions. An alternative approach can be utilized using beamforming technique. Basically what beamforming does is to channelize the energy in one direction, thus making sure that the signal is reaching farther areas. Also, after enabling beamforming mechanism, the signal strength available at a point in that 'beam' direction increases, thus increasing throughput at that point. At long ranges 802.11ac does not offer significant advantages over omnidirectional antennas. So, for 802.11ac, it is most efficient to allow clients to connect with 64-QAM rates (middle range of MCS rates from 0 to 9), from a farther distance. The device that does the beamforming is known as the beamformer and the one on the receiving end is known as the beamformee, and for such a topology, the data goes from the beamformer to the beamformee. In 802.11ac, beamforming gains are

expected to be approximately 3 to 4 dB in the direction of the ‘beam’. This will approximately correspond to an increase of one MCS rate for a mid-range transmission. The key advantage with transmit beamforming is the ability to significantly improve link performance to a low complexity device. This advantage is shown in Figure 3.6, which depicts a beamforming device with four antennas, typically an AP. The device at the other end of the link has only two antennas, typical of a small client device, like a smartphone. Such a system would benefit from 4×2 transmit beamforming gain from AP to smartphone. One of the most important

Figure 3.5 Advantage of beamforming [EPne]

differences between 802.11n and 802.11ac is that beamforming has been greatly simplified in 802.11ac. The 802.11n standard had multiple methods of beamforming and implementing them in hardware was not chosen by vendors, and hence some proprietary solutions were seen in the real world. However, 802.11ac has defined only one type of beamforming, explicit beamforming. It is normally done for the traffic from the AP (beamformer) to the client (beamformee), only in the downlink direction. This has advantages when using a MU-MIMO system considering the fact that the AP is a stationary entity that can have multiple antennas, and often the client (a smartphone) has 1 or 2 antennas at the most.

3.3.1 Null Data Packet Explicit Beamforming in 802.11ac

Beamforming depends on channel calibration procedures, called channel sounding [2], which is used to determine how to radiate energy in a preferred direction. Beamforming enables the endpoints at either side of a link to get maximum performance by taking advantage of channels that have strong performance while avoiding paths and carriers that have weak performance. First we will see how channel sounding in 802.11ac occurs with the NDPA, Null Data Packet (NDP) and the Beamforming Compressed Action frame. Then we will see the specifics of the structure of these frames and the underlying mechanisms that enable the computation of the various matrices involved.

General description of the sounding and calibration procedure [Gas13]:

1. The beamformer begins the process by transmitting the NDPA frame, which is used to gain control of the channel. All the beamformees listen to the NDPA, and will remain up to receive the NDP frame, while all other stations will simply defer channel access until the sounding sequence is complete.
2. The beamformer follows the NDPA with the NDP frame.
3. Once the NDP is received by a beamformee, each OFDM subcarrier in the training field of the NDP frame is processed independently in its own matrix that describes the performance of the subcarrier between each transmitter antenna element and each receiver antenna element. The contents of the matrix are based on the received signal power and phase shifts between each pair of antennas.
4. This matrix is transformed by a matrix multiplication operation, called a Givens rotation, which depends on parameters called 'angles'. Rather than transmitting the full feedback matrix, the beamformee calculates the angles based on the matrix rotation that results in compression of the size of the total data that is sent back to the beamformer.
5. This compressed feedback form is known as the feedback matrix (denoted by letter V in the 802.11ac standard), and is returned to the beamformer. Only one set of angles is required to summarize the radio link performance for all of the OFDM subcarriers,

though, the set of angles can be quite large with wider channels. When transmitting the feedback matrix, there are three main factors that determine its size. The size of the feedback matrix is directly proportional to them. They are:

- (a) Wider channels (because they have more OFDM subcarriers).
- (b) The number of pairwise combinations of transmitter and receiver antennas.
- (c) The resolution of the angles. The 802.11ac standard specifies two sizes in which the angles can be represented.

The 802.11ac standard also specifies the order in which these angles are transmitted.

6. In Single User (SU) beamforming, there is one feedback matrix sent from the beamformee and one steering matrix used. In Multi User (MU) beamforming, each beamformee sends a feedback matrix and the beamformer must maintain a steering matrix for each client.
7. After the beamformer receives the feedback matrix, it uses the feedback matrix to calculate the steering matrix (denoted by letter Q in the 802.11ac standard) for transmissions to the beamformee. Matrix operations allow the spatial mapper to alter the signal to be transmitted for each OFDM subcarrier over each path to the receiver in one operation. After applying the steering matrix to the data for transmission, it will leave the antenna array in a non-omnidirectional pattern thus creating a 'beam' towards the beamformee.

3.3.2 Frames used for Beamforming

3.3.2.1 Null Data Packet Announcement

The channel sounding procedure begins when a beamformer transmits the NDPA. The NDPA is a control frame (Figure 3.7).

Fields in the NDPA are:

1. Receiver's Address (RA): This field is set to MAC address of the beamformee for a SU beamforming client. For MU beamforming, it is set to broadcast address FF:FF:FF:FF:FF:FE

Figure 3.6 NDPA frame structure [Gas13]

2. **Duration:** The duration inserted in the NDPA is total time taken for the $T_{NDPA} + T_{SIFS} + T_{NDP} + T_{SIFS} + T_{BeamformingCompressedActionFrame}$ (for SU beamforming). For MU beamforming, it is $T_{NDPA} + T_{SIFS} + T_{NDP} + T_{SIFS} + (N_{Beamformees} - 1) \times T_{BeamformingReportPoll} + N_{Beamformees} \times T_{BeamformingCompressedActionFrame}$.
3. **Sounding Dialog Token:** This field acts as a token number used to identify the NDPA frame. This is matched with the token value in the received Beamforming Compressed Action Frame. It is used for recognizing delayed responses.
4. **Station (STA) Info:** For every beamformee there is one STA Info field. It has the following subfields:
 - (a) **AID12** - Contains the 12 least significant bits of the Association ID (AID) of a STA expected to process the following VHT NDP frame and prepare the sounding feedback. Equal to 0 if the STA is an AP, mesh STA or STA that is a member of an IBSS [2].
 - (b) **Feedback type:** Indicates the type of feedback requested. It is set to 0 for SU and 1 for MU.
 - (c) **N_c Index:** If the Feedback Type field indicates MU, then N_c Index indicates the number of columns, N_c , in the Compressed Beamforming Feedback Matrix subfield minus one [2]:

- i. Set to 0 to request $N_c = 1$
- ii. Set to 1 to request $N_c = 2$
- iii. ...
- iv. Set to 7 to request $N_c = 8$
- v. Reserved if the Feedback Type field indicates SU.

3.3.2.2 Null Data Packet

It is a PLCP frame with no data field since channel sounding can be carried out by analyzing the received training symbols in the PLCP header. Within an NDP frame (Figure 3.7) there is one VHT-LTF for each spatial stream used in transmission. The remaining fields are the same as the VHT format preamble as mentioned previously.

Figure 3.7 NDP frame structure [Gas13]

3.3.2.3 Beamforming Report Poll Frame

This frame (Figure 3.8) is exclusively used in Multi User Beamforming and is of control type. It is used by the beamformer to poll the beamformees for the Compressed Beamforming Action Frame. On receiving this frame, a beamformee sends the Compressed Beamforming Action Frame after SIFS interval.

Figure 3.8 Beamforming report poll frame structure [Gas13]

The fields in this frame are as follows:

1. RA (Receiver's Address) - Address of the intended recipient.
2. TA - Transmitter's Address.
3. Feedback Segment Retransmission Bitmap - Indicates the requested feedback segments of the Compressed Beamforming Report. If the bit in position n is 1 ($n = 0$ for LSB and $n = 1$ for MSB), then the feedback segment with the Remaining Feedback Segments subfield in the VHT MIMO Control field equal to n is requested.

3.3.2.4 Compressed Beamforming Action Frame

This is the frame that a beamformee sends to the beamformer with the feedback matrix in it. This is the matrix that the beamformer then uses to calculate the steering matrix. Figure 3.9 shows the structure of the compressed beamforming action frame for a Single User (SU) and, for Multi User (MU) it is shown by Figure 3.10.

Figure 3.9 SU Compressed beamforming action frame [Gas13]

The fields in the compressed beamforming action frame are:

1. Category: Describes the category of Action Frame. For VHT type action frame it is 21.

Figure 3.10 MU Compressed beamforming action frame [Gas13]

2. VHT Action: This field describes what kind of VHT Action Frame it is. For VHT Compressed Beamforming this value is 0.
3. VHT MIMO Control: This field is included in every Compressed Beamforming Report frame. It describes the parameters of the compressed beamforming report.
 - (a) N_c Index: Indicates the number of columns, N_c , in the compressed beamforming feedback matrix minus one:
 - i. Set to 0 for $N_c = 1$.
 - ii. Set to 1 for $N_c = 2$.
 - iii. ...
 - iv. Set to 7 for $N_c = 8$.
 - (b) N_r Index: Indicates the number of rows, N_r , in the compressed beamforming feedback matrix minus one:
 - i. Set to 0 for $N_r = 1$.
 - ii. Set to 1 for $N_r = 2$.
 - iii. ...
 - iv. Set to 7 for $N_r = 8$.
 - (c) Channel Width: Indicates the width of the channel in which the measurement to create the compressed beamforming feedback matrix was made:

- i. Set to 0 for 20 MHz.
 - ii. Set to 1 for 40 MHz.
 - iii. Set to 2 for 80 MHz.
 - iv. Set to 3 for 160 MHz or 80+80 MHz.
- (d) Grouping: Indicates the subcarrier grouping, N_g , used for the compressed beamforming feedback matrix:
 - i. Set to 0 for $N_g = 1$ (No grouping).
 - ii. Set to 1 for $N_g = 2$.
 - iii. Set to 2 for $N_g = 4$.
 - iv. The value 3 is reserved.
- (e) Codebook Information: Indicates the size of codebook entries:
 - i. If Feedback Type is SU:
 - A. Set to 0 for 2 bits for ψ , 4 bits for ϕ .
 - B. Set to 1 for 4 bits for ψ , 6 bits for ϕ .
 - ii. If Feedback Type is MU:
 - A. Set to 0 for 5 bits for ψ , 7 bits for ϕ .
 - B. Set to 1 for 7 bits for ψ , 9 bits for ϕ .
- (f) Feedback Type: Indicates the feedback type.
 - i. Set to 0 for SU.
 - ii. Set to 1 for MU.
- (g) Remaining Feedback Segments: Indicates the number of remaining feedback segments for the associated VHT Compressed Beamforming frame.
 - i. Set to 0 for the last feedback segment of a segmented report or the only feedback segment of an unsegmented report. (In a retransmitted feedback segment, the field is set to the same value associated with the feedback segment in the original transmission).
 - ii. Set to a value between 1 and 6 for a feedback segment that is neither the first nor the last of a segmented report.

- iii. Set to a value between 1 and 7 for a feedback segment that is not the last feedback segment of a segmented report.
- (h) First Feedback Segment:
 - i. Set to 1 for the first feedback segment of a segmented report or the only feedback segment of an unsegmented report.
 - ii. Set to 0 if it is not the first feedback segment or if the VHT Compressed Beamforming Report field and MU Exclusive Beamforming Report field are not present in the frame.
 - iii. In a retransmitted feedback segment, the field is set to the same value associated with the feedback segment in the original transmission.
- (i) Sounding Dialog Token Number: The sounding dialog token from the VHT NDPA frame.
- 4. Compressed Beamforming Report: This field's information is given in A.1.
- 5. MU Exclusive Beamforming Report Field: This field is only in the frames that are in MU scenario. This field's information is given in A.2.

3.3.3 Single User Beamforming

This is the most basic use of the beamforming mechanism where a single transmitter sends data to a single receiver. The exchange of packets that takes place for sounding is shown in Figure 3.11. Here, the beamformer sends the NDPA for notifying the beamformee (by including the AID (Association Identification) of that beamformee), that an NDP frame is going to follow. It sends the NDP frame after SIFS interval. The beamformee calculates the steering matrix and the coefficients are sent back in a compressed beamforming report. After this, the beamformer applies the angles to the spatial streams and sends the subsequent data packet. To constitute the case where a beamformee might be moving/mobile, the sounding of the channel is done every 30 ms [Gas13].

Figure 3.11 SU beamforming sounding sequence [Gas13].

3.3.4 Multi User Beamforming

This is an extension of the Single User (SU) Beamforming. As we can see in Figure 3.12, the procedure is similar to that of the SU beamforming. But here, after the first client has responded with a compressed beamforming report, the beamformer polls the next beamformee. The beamformee to which the poll is directed (specified by the AID in the frame), sends the compressed beamforming report. This continues till all the beamformees associated with the beamformer have sent their compressed beamforming report.

Figure 3.12 MU beamforming sounding sequence [Gas13].

3.3.5 Computation of Feedback Matrix

Figure 3.13 MIMO system with transmit beamforming [Gas13].

A generic MIMO system's (Figure 3.13) general description after performing transmit beamforming on each subcarrier in the frequency band gives us Eq. 3.1 (for every subcarrier) [EPne].

$$Y_{N_{RX}} = \sqrt{\frac{\rho}{N_{TX}}} H_{(N_{RX} \times N_{TX})} V_{N_{TX} \times N_{SS}} X_{N_{SS}} + Z_{N_{RX}} \quad (3.1)$$

where, X is the transmitted data with dimension N_{SS} (number of spatial streams) $\times 1$; V is the transmit weighing matrix with dimension N_{TX} (number of transmit antennas) $\times N_{SS}$; Y is the received signal with dimension N_{RX} (number of receive antennas) $\times 1$; H is the channel fading matrix with dimension $N_{RX} \times N_{TX}$; Z is the additive white Gaussian noise (AWGN) defined as Complex Normal (0, 1) with dimension $N_{RX} \times 1$; and ρ is the SNR. The matrix V is the transmit weighing matrix, and is the one which we want to calculate to apply the weights to the transmit streams. This process starts with decomposing the channel matrix H by using

Single Value Decomposition (used for factorization of a matrix) shown by Eq. 3.2 [EPne].

$$H_{(N \times M)} = U_{(N \times N)} S_{(N \times M)} V_{(M \times M)}^* \quad (3.2)$$

where, V and U are unitary matrices and S is a diagonal matrix of singular values and V^* is the Hermitian (complex conjugate transpose) of V . H , the channel matrix is calculated by the receiver by using the known transmitted VHT-LTF to create an estimate of the channel to equalize the symbols received. Non-compressed beamforming matrices may require a large number of bits to represent the complex values with limited quantization loss. If the technique used to compute the beamforming weights results in unitary matrices, polar coordinates may be used to reduce the number of bits required for beamforming weights feedback. To decompose the beamforming matrix V into polar values, we apply to it a sequence of Givens rotations to it [EPne]. The Given's rotation matrix is given by:

$$G_{li}(\psi) = \begin{pmatrix} I_{i-1} & 0 & 0 & 0 & 0 \\ 0 & \cos(\psi_{l,i}) & 0 & \sin(\psi_{l,i}) & 0 \\ 0 & 0 & I_{l-i-1} & 0 & 0 \\ 0 & -\sin(\psi_{l,i}) & 0 & \cos(\psi_{l,i}) & 0 \\ 0 & 0 & 0 & 0 & I_{M-l} \end{pmatrix} \quad (3.3)$$

where I_m is an $m \times n$ Identity matrix, and the terms $\cos(\psi_{l,i})$ and $\sin(\psi_{l,i})$ are located at row l column i . However, since the beamforming matrix V may comprise complex values, preprocessing steps are required before applying Givens rotations to the matrix V . A diagonal matrix D_i is derived for a matrix V such that the elements of column i of $D_i^* V$ are all non-negative real numbers, given by Eq. 3.4 [EPne].

$$D_i = \begin{pmatrix} I_{i-1} & 0 & \cdots & \cdots & 0 \\ 0 & e^{j\phi_{i,i}} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & e^{j\phi_{M-1,i}} & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (3.4)$$

The angles $\phi_{(l,i)}$ are computed as $\phi_{(l,i)} = \angle(V_{(l,i)})$. Since the last element of D_i is always

1, the elements in the last row of V are not altered. Therefore, a prior step of multiplying by $\tilde{D}$ is required, which is given in Eq. 3.5 [EPne].

$$\tilde{D} = \begin{pmatrix} e^{j\theta_1} & 0 & \dots & \dots & 0 \\ 0 & e^{j\theta_2} & 0 & \dots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & e^{j\theta_{N-1}} & 0 \\ 0 & 0 & \dots & 0 & e^{j\theta_N} \end{pmatrix} \quad (3.5)$$

where $\theta_i = \text{angle}(V_{M,i})$. Therefore an $M \times N$ beamforming matrix is decomposed into a sequence of $\tilde{D}$, D_i and $G_{li}(\psi_{l,i})$ given by the following Eq. 3.6 [EPne].

$$V = \left(\prod_{i=1}^{\min(N,M-1)} [D_i \prod_{l=i+1}^M G_{li}^T(\psi_{l,i})] \times \tilde{I}_{M \times N} \right) \tilde{D} \quad (3.6)$$

where $I_{(M \times N)}$ is an $M \times N$ identity matrix with extra columns and rows filled with zeros when $M \neq N$. At this point the explicit feedback angles $\phi_{(l,i)}$ and $\psi_{(l,i)}$ are transmitted from the beamformee to the beamformer. The beamformer computes a transmit beamforming weighting matrix $\tilde{V}$ for each subcarrier as follows (Eq. 3.7) [EPne]:

$$\tilde{V} = V \tilde{D}^* \quad (3.7)$$

After converting to polar coordinates, the angles are quantized. The angles ϕ are quantized between $0 - 2\pi$ and the angles ψ are quantized between $0 - \pi/2$ as given by the equations Eq. 3.8 and Eq. 3.9 [EPne].

$$\phi = \pi \left(\frac{1}{2^{b+2}} + \frac{k}{2^{b+2}} \right) \quad k = 0, 1, \dots, 2^{b+2} - 1 \quad (3.8)$$

$$\psi = \pi \left(\frac{1}{2^{b+2}} + \frac{k}{2^{b+2}} \right) \quad k = 0, 1, \dots, 2^b - 1 \quad (3.9)$$

where $(b+2)$ is the number of bits used to quantize ϕ and b is the number of bits used to quantize ψ . The total number of bits transmitted per subcarrier for the matrix V is given by the number of angles multiplied by the number of bits used for quantization. Grouping of

subcarriers may be used to further reduce the feedback overhead with compressed beamforming matrices. This information of whether to group the subcarriers and by how much, is given by the VHT MIMO control field.

For a MU-MIMO system, the computation for the steering matrix is slightly different. The beamformees report the ratio of the SNRs of the subcarriers in the MU Exclusive Beamforming Report Field. These are included in the steering matrix calculation, so that the beamformer can pre-code before sending the data. This is done to reduce the interference for two clients (beamformees) kept beside each other. The working of MU-MIMO depends on how good the Space Division Multiple Access (SDMA) works in separating the individual beams. The actual explanation is out of the scope of this thesis [EPne], but the end result is the result of weights which on application to transmission, allows the beamformer to transmit multiple frames simultaneously. An example of downlink data transmission for MU-MIMO

Figure 3.14 MU-MIMO data transmission [Gas13].

system is seen in Figure 3.14 [Gas13]. The beamformer applies the aforementioned weights that give rise to the MU Data transmission. The Block Ack Request frames (BAR) are sent by the beamformer to the beamformees and they reply with a Block Ack (BA) frame as an Acknowledgement for the receipt of the data frame. As we can see, MU-MIMO combines the transmission of various data frames and sends them in one 'frame time'. This improves the

throughput by a factor of up to the number of multiple users.

Chapter 4

Software Defined Radio, USRP2 and GNUradio

Software defined radios (SDRs) [Mit93] are a relatively new paradigm of the communication industry. SDRs basically put the ‘software’ part of a radio communication system as close to the antenna as possible. This is desirable since then different types of modulation and other signal processing transformations can be applied in the software. Hence, SDR can be programmed to suit the needs of an application. An ideal SDR receiver will get the analog signal from an antenna, convert it to digital and then pass it along to a Digital Signal Processor (DSP) for further processing. But, since such an ideal SDR would have infinite bandwidth, current implementations first convert the Radio Frequency (RF) to Intermediate Frequency (IF), which is then digitized, down-converted and then given to a DSP.

The Universal Software Radio Peripheral [LLCa], is a SDR designed to allow computers to function as high bandwidth software radios, created by Ettus Research LLC. It serves as a digital baseband and Intermediate Frequency (IF) section of a radio communication system and modulation and demodulation of the signal is done on the machine’s CPU. Only the high-speed general-purpose operations like Digital Up Conversion (DUC) and Digital Down Conversion (DDC) are done on the Field Programmable Gate Array (FPGA) residing on the USRP2. As mentioned before, all the other complex baseband signal handling is done on a general purpose computer. The USRP2 platform has the following features [LLCa]:

1. A Xilinx Spartan 3-2000 FPGA.
2. Gigabit Ethernet interface.
3. Two 100 Million Samples/sec, 14-bit, analog-to-digital converters, LTC2284, 72.4 dB SNR and 85 dB SFDR for signals at the Nyquist frequency.
4. Two 400 Million Samples/sec, 16-bit, digital-to-analog converters, AD9777. 160 Million Samples/sec without interpolation, up to 400 Million Samples/sec with $8\times$ interpolation.
5. Secure Digital (SD) card reader.

The USRP2 that we have used has a XCVR 2450 daughterboard [LLCb] installed on it that acts as a RF front end. It has Dual-band Transceiver capabilities with 100+mW output at 2.4-2.5 GHz and 50+mW output at 4.9-5.85 GHz.

A large community of developers and users have created and contributed to a very large code base of drivers and compatible software. The most well known example is GNURadio [Blo]. GNURadio is the software end that runs on the host computer for an SDR connected to that computer. This allows a user to create radio receivers and transmitters and other modules using signal processing blocks in GNURadio with the SDR as the RF end. These blocks are designed in C++ and Python, and are executed on a C++ engine for efficiency. GNURadio can also run in a standalone mode where a user can create waveforms, modulate them, analyze them, in a simulation like environment. A GNURadio Companion (GRC) user interface front end is also provided for easily connecting signal processing blocks by a drag-and-drop method. Using GRC really shows the user the modularity with which the GNURadio was designed.

We have used two USRP2 SDRs along with GNURadio running on a host PC machine to create our own ‘jammer’. One USRP2 serves as a receiver and recognizes the trigger NDPA packet that we use for jamming NDP frame, and the other USRP2 is designed to jam.

Chapter 5

Jamming Attacks using Beamforming

Beamforming is essentially used to increase the SNR at the beamformee that results in lower Bit Error Rate (BER), and hence a higher modulation rate can be used to send data to the beamformee, thus increasing overall throughput. As a mechanism, it is vulnerable to jamming attacks, for two reasons. First, the sounding consists of a PHY layer frame, called the NDP frame. Since this frame is used for calibration of the channel to derive the channel matrix at the beamformee, there is no Frame Check Sequence (FCS) equivalent to detect flipped bits. Second, the control (NDPA and Beamforming Report Poll), and management (Compressed Beamforming Report) overhead do not have any encryption. The NDPA control packet is sent out at a legacy rate so that all the clients (legacy and otherwise) can create a NAV and defer any transmissions. The Beamforming Compressed Report frame is of the management type, but even with the introduction of 802.11w [11], it has not been encompassed under this protocol. All of this has lead to the exposure of the vulnerabilities that are now listed:

1. Jamming an NDPA frame will essentially stop the beamforming mechanism from ever taking place. But, since an intelligent jammer requires a trigger to detect when the sounding is taking place, this frame can be used in a better way as a trigger to jam other packets in this sounding sequence.
2. Jamming the NDP frame might be the most effective since it is used for multiple clients, and just by jamming/scrambling the bits in this frame, multiple clients/beamformees in a MU-MIMO scenario will be affected as well as a single client.

3. Although jamming the beamforming compressed report would be effective in a single user scenario, it is not as efficient in a MU-MIMO scenario, since a jammer would have to jam every beamforming compressed action frame sent by each and every one of the beamformees.
4. The NDPA frame contains a duration field, which gives the time required for the entire sounding sequence. Spoofing such a frame with large duration field values will set large NAV vectors in other clients, thus creating a denial of service of the medium.
5. Since the compressed beamforming action frame is not included under the replay protection mechanism found in 802.11w[11], it can be captured and replayed. This type of attack will be most useful when the beamformee is mobile and moving, since the stale 'angle' values will train the beamformer antennas in an 'old' direction, thus reducing the throughput from the beamformer to the beamformee. This attack has the potential to reduce the throughput dramatically since the beamformer will direct the signal to the wrong direction. It is also possible to have a stationary beamformee and to inject a carefully crafted compressed beamforming report frame such that the beam is badly misdirected.

We are going to jam the NDP frames because this attack will affect all the clients in the network (as opposed to the beamforming compressed report replay attack). We will go ahead and show with a proof-of-concept setup that such a jamming attack on the NDP frame considerably reduces throughput in a network.

Chapter 6

Setup and Implementation

6.1 Setup

At the outset of this work, the official 802.11ac standard had not been released, and hence only a couple of 802.11ac draft compliant hardware devices were available in the market. For the same reason, as far as we know, no simulators for 802.11ac existed which we could have leveraged for getting results. At this point, Broadcom Corporation graciously agreed to let us have an 802.11ac AP for the thesis work. Since they fabricate their own 802.11ac cards and sell them to vendors, we were able to secure an 802.11ac reference board AP from them. Along with this, we also got a commercially available Netgear (A6200) card [Net] which we used as a client. The Netgear client was chosen since it is a commercially available, so that our implementation would be largely platform agnostic and would highlight the weaknesses of the 802.11ac standard. This setup facilitated a real world implementation, which is preferred to a simulation. A picture of the setup is shown in Figure 6.1

To clarify, Figure 6.2 shows the schematic of our setup. The GNURadio software runs on the host machine and controls the USRP2s. A machine which acts as a wireless sniffer has also been setup. This machine is used for creating packet traces of the wireless environment by sniffing in monitor mode. An 802.11ac AP is connected to the wired interface of the Wireshark sniffer via Ethernet. Similarly, two USRP2s are connected to the host machine via Ethernet. The Universal Software Radio Peripheral [LLCa], is a SDR designed to allow computers to

Figure 6.1 The experimental setup

function as high bandwidth software radios, created by Ettus Research LLC. It serves as a digital baseband and Intermediate Frequency (IF) section of a radio communication system and modulation and demodulation of the signal is done on the machine's CPU. Only the high-speed general-purpose operations like Digital Up Conversion (DUC) and Digital Down Conversion (DDC) are done on the Field Programmable Gate Array (FPGA) residing on the USRP2. As mentioned before, all the other complex baseband signal handling is done on a general purpose computer. The USRP2 platform has the following features [LLCa]:

1. A Xilinx Spartan 3-2000 FPGA.
2. Gigabit Ethernet interface.
3. Two 100 Million Samples/sec, 14-bit, analog-to-digital converters, LTC2284, 72.4 dB SNR and 85 dB SFDR for signals at the Nyquist frequency.
4. Two 400 Million Samples/sec, 16-bit, digital-to-analog converters, AD9777. 160 Million Samples/sec without interpolation, up to 400 Million Samples/sec with $8\times$ interpolation.
5. Secure Digital (SD) card reader.

The USRP2 that we have used has a XCVR 2450 daughterboard [LLCb] installed on it that acts as a RF front end. It has Dual-band Transceiver capabilities with 100+mW output at 2.4-2.5 GHz and 50+mW output at 4.9-5.85 GHz. We have used two USRP2 SDRs along with

Figure 6.2 Setup schematic

GNURadio running on a host PC machine to create our own ‘jammer’. One USRP2 serves as a receiver and recognizes the trigger NDPA packet that we use for jamming NDP frame, and the other USRP2 is designed to jam. The client is a laptop fitted with the Netgear A6200 card via USB.

The correct placement of all the equipment is very important to the working of the experiment. The jammer USRP2 is kept near the beamformee (client + netgear card) with a low transmit power, so as not to disturb the reception of the USRP2 sniffer kept near the 802.11ac AP. The client itself is kept at a point where the received signal from the AP is just about strong enough to pass MCS8 rate traffic. Since the LAN cable and the power cable of the jammer USRP2 is not very long we cannot put the AP and the client very far apart. Hence to simulate the effect of a long distance between the AP and the client, the transmit power is reduced on both.

6.2 USRP2 software setup

The host machine was prepared with a version of Xubuntu 12.04. We have specifically used this Linux distribution, since primarily we required a base Ubuntu OS for installing the USRP Hardware Driver (UHD), and also because 12.04 was the latest Long Term Support (LTS) version of Ubuntu with Xfce desktop (to make it lightweight). This was followed by installing the prerequisites and dependencies for the UHD, which was followed by installing the UHD itself (version 003.005.004). The UHD is used by the host machine to talk to USRP2s over the Ethernet and provides host APIs for the USRP2s. It is compatible with GNURadio [Blo]. After the installation of UHD, we then installed GNURadio version 3.7.1-52 [Blo]. Now, we had in place the software that runs on the host machine. The next step we did was to burn the SD card that comes with the USRP2s, which is used to install the FPGA and firmware images onto a USRP2. From the software installation point of view we were set on all the fronts. To get the optimum number of vector calculations per second from the host machine we have also installed and integrated Vector Optimized Library of Kernels (VOLK) [TR12] as suggested by the creators of GNURadio. This gives a huge boost in performance by using Single Instruction Multiple Data (SIMD) instructions, which is useful in speeding up signal processing applications. We have also tuned the network stack buffers of the host machine by changing the size of the parameters 'rmem-max' and 'wmem-max' to a much larger size, to handle the large number of complex baseband samples sent and received over the Ethernet to/from the GNURadio. This completes the host machine and USRP2 part of installation.

6.3 Access Point setup

We received the Broadcom Corporation AP with a preinstalled binary of their firmware and driver (version 6.37.15). This is a general release version, and hence available with the buyers of the Broadcom Corporation's chipsets and consequently is installed on the chipsets inside devices available on the market. Although the AP can be set up on 80 Mhz channels, the USRP2s are not able to handle more than 20 Mhz of bandwidth. Hence we put the AP on 20 Mhz. The other settings include setting only the 6 Mbps rate as the 'basic' legacy rate, so that all the control and management frames are sent at that rate. This enables the USRP2 to

successfully receive almost all the packets. We have also reduced the unnecessary traffic sent by the AP on the wireless side by disabling Spanning Tree Protocol (STP). After this we setup a basic service set on the AP.

6.4 Wireshark sniffer setup

The Wireshark sniffer machine has been setup with a 802.11ac card we got from Broadcom Corporation. We have installed that card on a MiniPCI Express to PCI adapter that lets us mount the card on the motherboard of the sniffer machine. After that, we installed Linux (Fedora core 15) along with the driver (provided by Broadcom Corporation) on the machine after which we installed the latest Wireshark version. The key function of this machine is that it is able to sniff and decode 802.11ac frames.

6.5 Client setup

The client is a Windows 7 machine with the Netgear A6200 card [Net] installed on it. We have disabled many unnecessary traffic generating services in Windows 7 like the IPv6 stack along with NetBIOS and File and Printer sharing service. We connect the client to the basic service set which is set up on the AP.

6.6 Implementation

For jamming the NDP frame, the basic piece of knowledge we require is when it will be transmitted. Since it is transmitted by the beamformer a SIFS interval after NDPA frame, we designed a jammer system that could detect the NDPA frame. First, we require an OFDM receiver capable of demodulating and decoding a 20 Mhz signal with 64 subcarriers, each spaced 312.5 kHz apart. After searching long and hard, we finally found an OFDM receiver implemented in GNURadio that runs on USRP2 [Blo13]. On installing and running it, we realized that it was not able to receive and decode all the packets it sniffed because it was a software implementation of a OFDM receiver. Hence, we removed all the extra wireless traffic (as mentioned before) and also chose channel 165 (5.825 Ghz) as there was no background

traffic on that channel. The implementation of the OFDM receiver by the authors in [Blo13] is seen in Figure 6.3. Figure 6.3 shows the interconnection of the various signal processing

Figure 6.3 802.11 a/g OFDM receiver for GNURadio [Blo13]

blocks in an OFDM receiver. Our first design of the jammer based on this receiver, decoded the MAC header and, on realizing that the frame was an NDPA frame, sent a jamming signal to jam the NDP frame. This approach was too slow, since GNURadio creates threads for every block and consequently, with more blocks, the responsiveness of the jammer was slow, sluggish and often missed the NDP frames. Therefore, we implemented a jammer as seen in the Figure 6.4. We then realized that decoding the MAC header was not required since we could determine the frame being sent based on the length subfield in the SIGNAL field of the PLCP header. Therefore, we removed the ‘OFDM Decode MAC’, ‘OFDM Parse MAC’ and ‘Socket PDU’ blocks from the OFDM receiver. We first verified that in the current implementation there was no other frame being sent on the same channel having the same length as the NDPA. Once verified, we built a jammer as shown in Figure 6.4. Figure 6.4 shows the full flow graph of the jammer whereas Figure 6.5 shows only the jammer portion of the whole implementation for better readability. The main function of the OFDM Decode Signal block (Figure 6.3) is to recognize the SIGNAL field from the PLCP header (since it is only a single OFDM symbol), and extract the length and modulation information of the proceeding

Figure 6.4 Full flow graph of jammer implementation in GNURadio

OFDM symbols of the rest of the frame. This block processes one OFDM symbol at a time. The ‘trigger block’ (Figure 6.5) is basically a modified OFDM Decode Signal block such that it outputs a ‘1’ if it detects an NDPA frame (based on the length and rate, which is 6 Mbps), and for all other symbols, it outputs a ‘0’. With this implementation, we have a trigger mechanism that recognizes the NDPA frame being put on the channel. The next part is to generate the jamming signal. We do this by first repeating the ‘1’s and ‘0’s (output of the ‘trigger block’) 512 times each. (This number had to be a multiple of 64 since we do a 64 point Fourier Transform as we will see later.) We ‘AND’ these with a random signal to generate a stream which has 512

Figure 6.5 Jammer part of the entire implementation

random numbers and the rest are '0's which are given as an input to 'Int to Float' block that converts the incoming integer values to floating point values. These are then fed into a 'Float to Complex' block which takes real and imaginary streams as input and generates complex numbers by combining them.

Next, these are fed to a 64 point Inverse Fast Fourier Transform (IFFT) block which outputs 64 samples. This corresponds to the 64 subcarriers used in the OFDM transmission and reception in 802.11a. Next, these are fed to a ‘constant multiplier’ block that multiplies every input sample with ‘10000’. This is done to increase the amplitude of the signal. This block is necessary, since, as we previously mentioned, the transmit gain of the jammer USRP2 has to be such that the sniffer USRP2 is not compromised in receiving and decoding the OFDM symbols. The IFFT block essentially generates samples of zero magnitude when it does an IFFT on 64 samples, each having a zero magnitude. Hence, the constant multiplier block output also has zero magnitude only for such samples. This stream of numbers is then passed onto an interpolation block which is setup to generate 10^8 output samples per input sample. This is then fed to the transmitter USRP2 block that interpolates the samples as

required to get a rate of 20 Msps and then transmits on the 165 channel. The values for the repeat block (i.e. 512), and the interpolation block (10×10^{-9}) are related to one another. We have reached these numbers empirically, since lowering the number of samples sent to the transmitter USRP2 block gives an ‘Underflow’ message, while sending more than it can handle gives the ‘Overflow’ message. We also inserted a delay block after the ‘trigger block’ to make sure that the jamming signal being sent overlaps the exact transmission of the NDP frame without jamming other frames. After experimenting with different values of the delay block, we realized that the best reading was achieved by setting the delay as zero, i.e. no delay.

After setting up the jammer, we started downlink traffic by simply pinging the wireless client from the AP. After much testing and tweaking we realized that Internet Control Message Protocol (ICMP) packets are not beamformed by the AP. For this phenomenon, we reasoned that since ICMP packets should reach the client because they are sent with the highest priority, the AP does not do any beamforming and sends them at a legacy rate, so that the client definitely gets the packet in the first transmission itself (i.e. the number of retransmissions are zero). Hence, we needed a traffic generating tool. We used ‘iperf’ and it was installed on both the client (beamformee) and the wired interface of the sniffer machine. We generated User Datagram Protocol (UDP) traffic that went from the wired interface of the sniffer machine to the AP, which relayed the packets on the wireless medium to the client.

6.7 Expected results

We have generated the wireless sniffer traces used for the observations in the following way. First, we just generate the downlink traffic, sniff it and save the trace. Then we turn on the jammers and again sniff the medium and save the trace. We expect that the NDP frames will be jammed enough so that the client (beamformee) will read them and infer incorrect angles, which it then sends back to the AP (beamformer) in the compressed beamforming action frame. The AP will use these angles, generate the steering matrix and steer the ‘beam’ in a different direction than what would have been generated if the jamming had not taken place. This will reduce the effective SNR of the AP’s power at the client, thus making the AP reduce its MCS rate. Over time this will significantly affect the downlink throughput. Hence we can say that the jamming has been effective.

Chapter 7

Readings and Observation

First we will see how enabling and disabling beamforming affects the rate of the packets sent from the beamformer to the beamformee. We performed a test with the AP sending downlink traffic to the client generated using iperf on the sniffer machine's wired interface [Figure 6.2]. Table 7.1 shows the rates at which the packets are sent and received with and without beamforming. This table has been generated from the information retrieved on the AP console and shows the number of transmitted packets for every MCS rate from the AP's perspective. When there is no beamforming, 59.86% of the packets sent, are sent on MCS 7, 39.96% packets are sent at MCS 8 rate and only 0.18% packets are sent at MCS 9. When beamforming is turned on, the picture changes completely. This time, 99.79% of the packets are sent at MCS 9 and 0.21% packets are sent at MCS 8. This increase in MCS rate is due to higher SNR and consequently lower BER. It does not depend upon the packet size or

Table 7.1 Comparison of transmitted packets with and without beamforming

802.11ac MCS rate	Beamforming OFF (number of transmitted packets)	Beamforming ON (number of transmitted packets)
MCS 7	36606	0
MCS 8	24434	199
MCS 9	111	96539

aggregation or any other parameter. (Please note that since we needed to prove the increase in throughput using beamforming, and not test the jamming mechanism, we have made sure that the restrictions previously mentioned on the client and the AP (like channel bandwidth of 20 Mhz), have been removed. Also, the number shown in the table above is not only of the UDP packets, but of all the packets the AP beamformed.) The throughput shown by the iperf software is 20 Mbps without beamforming and 35 Mbps with beamforming. We can clearly see a large increase in throughput when beamforming is turned ON, even with the beamforming overhead. This overhead is pretty small for a single client sounding sequence, about 1% of the airtime [Gas13]. But as the number of clients increase, the sounding sequence overhead increases linearly. This is where the MU-MIMO system excels by sending data downlink to multiple clients in the same frame ‘air time’.

Now, let us compare the readings with and without jamming. Table 7.2 shows the readings for the number of packets transferred at 802.11ac and 802.11n rates. (Please note that since we had to use the USRP2s, the previous restrictions of channel bandwidth to 20 Mhz have been enforced again). Two trials of the experiment were performed on different days and

Table 7.2 Comparison of number of packets sent at various MCS rates with and without jamming

MCS Rate	Without jamming (tx packets)		With jamming (tx packets)	
	Trial 1	Trial 2	Trial 1	Trial 2
802.11ac MCS 8	913	824	576	205
802.11ac MCS 7	0	271	0	747
802.11n MCS 8	0	0	327	10

different times. Let us consider the example of ‘Trial 1’. For ‘Trial 1’, Table 7.2 clearly shows that when there is no jamming taking place, all the packets are sent at 802.11ac MCS 8 (the highest rate possible with 20 Mhz channel bandwidth). When jamming takes place, some of the packets (63.78%) are still sent at 802.11ac MCS 8 rate while the others (36.22%) are sent at 802.11n MCS 8. (IMPORTANT: this number reflects not only the beamformed UDP packets,

but all packets that are getting beamformed by the AP).

Table 7.3 shows the number of UDP packets (going downlink from the AP to the client) being beamformed when compared to the total number of UDP packets sent. As seen in Table

Table 7.3 UDP frames beamformed (not jammed v/s jammed)

	Without jamming (# of frames)		With jamming (# of frames)	
	Trial 1	Trial 2	Trial 1	Trial 2
UDP total transmitted	713	918	635	837
UDP beamformed	709	890	43	107
UDP retried frames	5	18	16	37

7.3, for ‘Trial 1’, without jamming, 99.43% of the UDP packets sent by the AP are beamformed whereas only 6.77% of the packets sent with jamming turned ON are being beamformed. This should ideally be 0% but due to the imperfect synchronization of the processes being scheduled on the host machine, which runs the jammer, we see some packets being sent using beamforming. It is also possible that jamming a few NDP frames did not change the ‘angles’ enough to destroy beamforming. We can see that the number of retried UDP packets are not much different in both scenarios. This assures us that the jamming signal did not ‘step over’ or jam the UDP packets themselves. Thus, we can say that the AP was not forced to return to other modulation scheme because the transmission of the previous packet failed. Also, on examining the traces, we can see that the Compressed Beamforming Action frame sent by the beamformee to the beamformer also does not show any retransmissions, thus confirming that the jamming signal is indeed jamming only the NDP frames. The difference in the readings when comparing Table 7.2 and Table 7.3 is due to the fact that the Received Signal Strength Indicator (RSSI) of the beamformer at beamformee varies. This, coupled with ‘the not perfectly synchronized jammer’, is giving rise to the discrepancy. In order to show jamming accounts for the change in the MCS rates as well as the UDP packets not getting beamformed, we now verify that the NDP frame was getting jammed. This can be

done by dissecting the Compressed Beamforming Action frames sent by the beamformee to the beamformer. Since we have captured the traces, we analyzed them and came up with the following facts about the AP-client connection. We verified these by cross checking with the specifications and the AP configuration. The AP transmitted on three antennas and the Netgear card received on two, with two spatial streams. Therefore, the feedback matrix for every subcarrier was $N_r \times N_c = 3 \times 2$ with the number of angles $N_a = 6$ (Figure A.2). This gave us the angles $\phi_{11}, \phi_{21}, \psi_{21}, \psi_{31}, \phi_{22}$ and ψ_{32} inserted in that order in the packet for 52 subcarriers. By examining the VHT MIMO control field, we can see that every ϕ angle is described by using 6 bits. This means that we can have ($2^6 = 64$) different quantized values for ϕ which lie between 0 to 2π . Similarly, ψ is described using 4 bits and hence can take ($2^4 = 16$) different values from 0 to $\pi/2$.

Let us consider, as an example, the ϕ_{21} angle. After obtaining all the compressed beamforming action frames, we scan through them to obtain the ϕ_{21} values for every subcarrier for every packet. These are then aggregated for each subcarrier for all packets. At this stage, we have groups of data, with each group corresponding to one subcarrier, and the data points in that group are the ϕ_{21} values for every packet. Then we take the mean and the standard deviation for every group giving us with 52 mean values and 52 standard deviation values, one for each carrier for ϕ_{21} . (This explanation is for ϕ_{21} , but each angle mentioned above has its own 52 sets of values for each subcarrier). This method is repeated for the trace generated before jamming and the one generated during jamming. Fig. 7.1 shows the graphs drawn from that data. For all the graphs, the X-axis represents the OFDM subcarriers. In Figure 7.1(a), the values plotted are the average values of ϕ_{21} generated from the trace before jamming. Figure 7.1(b) shows the average values of ϕ_{21} taken from the trace during jamming. The black line in all the four sub-graphs show a 26 point moving average taken across all the readings in that particular graph. This is drawn just to show how the mean progresses across all the readings so as to get an idea about the trend of the mean of that graph. Figure 7.1(c) shows the difference between values of every subcarrier in graphs given by Figures 7.1(a) and 7.1(b). This indicates that there is significant difference between the mean of ϕ_{21} angle values before and during jamming. This is shown in a more visually convincing way in the Figure 7.1(d), which shows the absolute difference between the mean of ϕ_{21} before and during jamming. In this figure, we can see the 26 point moving average to

Figure 7.1 Graphs for ϕ_{21} angle

be about 8, which means that the value of ϕ_{21} has changed by a margin of $8/64$ (since ϕ_{21} can be quantized to 64 different values), on an average for every subcarrier. For ψ_{21} , similar conclusions can also be drawn from graphs shown in Figure 7.2. This proves that jamming is indeed changing the preamble values in the NDP frame received by the beamformee enough to generally make the beamforming worthless. Similar points can be put forth for the remaining angles, viz., ϕ_{11} , ψ_{31} , ϕ_{22} and ψ_{32} . For these angles, the graphs which show the absolute difference in the mean before and during jamming can be seen in Figure 7.3.

Figure 7.2 Graphs for ψ_{21} angle

They also show similar results. Now, we prove the consistency of the results of jamming. More specifically, we now show that across all the packets collected in the trace, the values of the angles change in consistent way, and the mean, which we showed in previous graphs, was consistent. For this, we have split the packet capture into four groups and calculated the mean and standard deviation for each group as before. For ϕ_{21} , Figure 7.4 shows the difference between the mean (same as before), and Figure 7.5 shows the difference between the standard deviation for the readings of ϕ_{21} values for every group. In figures 7.4 and

Figure 7.3 Graphs for the absolute difference between averages for $\phi 11$, $\psi 31$, $\phi 22$ and $\psi 32$

7.5, the values shown by the 26 point average lines are the means across the graphs for each of the four groups of packets. As we can see, the four 26 point average lines in one graph are consistent in behavior with one another. From this we can ascertain the fact that all the angles sent in the compressed beamforming action frames are similar and consistent. Graphs with similar characteristics can be seen for other angles as well. The conclusion from all of these graphs coupled with the number of UDP packets not getting beamformed due to

Figure 7.4 Difference between the mean of ϕ_{21} before and during jamming when packets are split up

Figure 7.5 Difference between the standard deviation of ϕ_{21} before and during jamming when packets are split up

jamming, creates a compelling evidence that jamming the NDP frame has indeed almost completely stopped the beamforming mechanism. The experimental results show that jamming the NDP frame reduces the throughput because the AP does not beamform them at all, or it sends the data packets by beamforming at a lower MCS rate. This is almost equivalent to turning off the beamforming feature itself. As seen before, the decrease in throughput after turning off beamforming is from 35 Mbps to 20 Mbps. On calculating the percentage decrease in throughput, we can see that on an average the throughput decreases by 42.85%. The results at NCSU were replicated at Broadcom Corporation and the decrease there comes close to our results with a decrease of 31.6%. Let us see how this effect is compounded for a MU-MIMO system. In Figure 7.6, for three clients associated to an AP, the AP sends a data

Figure 7.6 Beamformed data transmission for a MU-MIMO system [Gas13]

frame to each one of them in one 'time frame'. Since MU-MIMO beamformed transmissions entirely depend on beamforming, jamming the NDP frame will result in the AP sending the data frames to each client sequentially since it cannot beamform and do simultaneous transmissions. Thus the data transmissions will take approximately three times longer than if beamforming to individual clients were used. This implies the throughput will decrease to 1/3rd of the beamformed throughput. This means that jamming the NDP frame will be very effective in reducing the throughput in a MU-MIMO scenario. This has even more effect with more clients connected to an AP (maximum of four [Gas13]). Consider four 2×2 clients associated with the AP. In this case, jamming the NDP frames will increase the time taken

by the data frames fourfold thus decreasing the throughput to a quarter of its true potential. This is in addition to the fact that all of the data will be transmitted at a lower rate.

Chapter 8

Conclusion and Future Work

We have created a proof-of-concept setup for exploiting a vulnerability that exists in the beamforming mechanism in the 802.11ac standard. A standardized method for beamforming has been introduced in the 802.11ac standard known as explicit beamforming. This has been done to implement MU-MIMO topologies to increase the overall downlink throughput from the AP to the clients. Since MU-MIMO relies on the working of beamforming mechanism, jamming the mechanism effectively stops the MU-MIMO beamformed transmissions, reducing the throughput in the network to approximately $1/n^{th}$ of beamformed users. Also, this jamming will be very difficult to detect since it is infrequent, always transmitted during other traffic, and can be different on each transmission which makes it very effective.

This vulnerability can be mitigated by not using explicit sounding mechanism and using the preambles in the data frames themselves for calculating the feedback matrix. This matrix can be returned to the beamformer by piggybacking onto some other frame. Implementation of this kind of mitigation is left as a part of future work.

The direction in which the 802.11 standards are getting drafted, implies that the beamforming mechanism and MU-MIMO are here to stay and will play an important role in increasing the throughput in future wireless deployments. 802.11ad is the standard for Giga-bit throughput speed in 60 Ghz band which uses beamforming for increasing throughput. This is also seen in 802.11af standard which brings WLAN operation in TV spectrum (Ultra High Frequency and Very High Frequency bands). Our work is very fundamental in its

approach since we feel that beamforming is the key to improving throughput in the future deployments, because it brings to WLANs the concept of Spatial Division Multiplexing (SDM), which directly resembles switching in wired networks.

Since the software defined radios are limited by their performance with respect to bandwidth limitations and performance issues, we hope that this proof-of-concept is a stepping-stone to something like finding mitigation mechanisms for the attacks we have presented here.

BIBLIOGRAPHY

- [BS03] Bellardo, J. & Savage, S. “802.11 Denial-of-service Attacks: Real Vulnerabilities and Practical Solutions”. *Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12*. SSYM’03. Washington, DC: USENIX Association, 2003, pp. 2–2.
- [Blo13] Bloessl, B. et al. “An IEEE 802.11a/G/P OFDM Receiver for GNU Radio”. *Proceedings of the Second Workshop on Software Radio Implementation Forum*. SRIF ’13. Hong Kong, China: ACM, 2013, pp. 9–16.
- [Blo] Blossom, E. *GNURadio*. GNU project.
- [EPne] Eldad Perahia, R. S. *Next Generation Wireless LANs: 802.11n and 802.11ac*. Cambridge, June 2013.
- [Flu01] Fluhrer, S. R. et al. “Weaknesses in the Key Scheduling Algorithm of RC4”. *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*. SAC ’01. London, UK, UK: Springer-Verlag, 2001, pp. 1–24.
- [Gas13] Gast, M. *802.11ac: A Survival Guide*. O’REILLY, 2013.
- [11] “IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Protected Management Frames”. *IEEE Std 802.11w-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, and IEEE Std 802.11y-2008)* (2009), pp. 1–111.
- [12] “IEEE Standard for Information technology–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements”. *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003))* (2005), pp. 1–212.
- [2] “IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 4: Enhancements for Very High Throughput for operation in bands below 6GHz”. *IEEE Std 802.11ac* (2014).
- [1] “IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific require-

- ments Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)* (2012), pp. 1–2793.
- [KV03] Kyasanur, P. & Vaidya, N. "Detection and handling of MAC layer misbehavior in wireless networks". *Dependable Systems and Networks, 2003. Proceedings. 2003 International Conference on.* 2003, pp. 173–182.
- [LLCa] LLC, E. R. *Universal Software Radio Peripheral 2*. Ettus Research LLC.
- [LLCb] LLC, E. R. *XCVR 2450 daughterboard manual*. Ettus Research LLC.
- [Mit93] Mitola J., I. "Software radios: Survey, critical evaluation and future directions". *Aerospace and Electronic Systems Magazine, IEEE* **8.4** (1993), pp. 25–36.
- [Net] Netgear. *Netgear A6200 Wifi USB Adapter Data Sheet*. Netgear. 350E.PlumeriaDrive, SanJose, CA95134-1911 USA.
- [Pel11] Pelechrinis, K. et al. "Denial of Service Attacks in Wireless Networks: The Case of Jammers". *Communications Surveys Tutorials, IEEE* **13.2** (2011), pp. 245–257.
- [TR12] T. Rondeau N. McCarthy, T. O. "SIMD Programming in GNU Radio: Maintainable and User-Friendly Algorithm Optimization with VOLK". *Proceedings of SDR 2012*. SDR 2012. Brussels, Belgium, 2012.
- [TA06] Thuente, D. J. & Acharya, M. "Intelligent Jamming in Wireless Networks with Applications to 802.11B and Other Networks". *Proceedings of the 2006 IEEE Conference on Military Communications*. MILCOM'06. Washington, D.C.: IEEE Press, 2006, pp. 1075–1081.
- [Xu05] Xu, W. et al. "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks". *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '05. Urbana-Champaign, IL, USA: ACM, 2005, pp. 46–57.

APPENDIX

Appendix A

Reference Tables

A.1 Compressed Beamforming Report

This field contains all the feedback angles sent from the beamformee to the beamformer. Its structure [Figure A.1] is shown as it appears in the 802.11ac v7.0 draft [2]. N_a is the number of angles used for beamforming given by the Figure A.2. This figure shows the entries in the table which describes the order of the angles in the feedback subfield. Although not entirely shown, it gives an idea of the way the angles are ordered. b_ϕ and b_ψ are the number of bits used to define the ϕ and ψ angles respectively given from the MIMO control field. The number of subcarriers for which the angles are calculated are given in Figure A.3. This figure shows a part of the table which describes the subcarriers for different channel widths. The average SNR is given by the table in the Figure A.4

Field	Size (bits)	Meaning
Average SNR of Space-Time Stream 1	8	Signal-to-noise ratio at the beamformer for space-time stream 1 averaged over all data subcarriers. See Table 8-53h (Average SNR of Space-Time Stream i subfield).
...	...	...
Average SNR of Space-Time Stream N_c	8	Signal-to-noise ratio at the beamformer for space-time stream N_c averaged over all data subcarriers. See Table 8-53h (Average SNR of Space-Time Stream i subfield).
Compressed Beamforming Feedback Matrix V for subcarrier $k = scidx(0)$	$Na \times (b_\psi + b_\phi)/2$	Compressed beamforming feedback matrix as defined in Table 8-53d (Order of angles in the Compressed Beamforming Feedback Matrix subfield)
Compressed Beamforming Feedback Matrix V for subcarrier $k = scidx(1)$	$Na \times (b_\psi + b_\phi)/2$	Compressed beamforming feedback matrix as defined in Table 8-53d (Order of angles in the Compressed Beamforming Feedback Matrix subfield)
Compressed Beamforming Feedback Matrix V for subcarrier $k = scidx(2)$	$Na \times (b_\psi + b_\phi)/2$	Compressed beamforming feedback matrix as defined in Table 8-53d (Order of angles in the Compressed Beamforming Feedback Matrix subfield)
...	...	...
Compressed Beamforming Feedback Matrix V for subcarrier $k = scidx(N_s - 1)$	$Na \times (b_\psi + b_\phi)/2$	Compressed beamforming feedback matrix as defined in Table 8-53d (Order of angles in the Compressed Beamforming Feedback Matrix subfield)
NOTE— $scidx()$ is defined in Table 8-53g (Subcarriers for which a Compressed Beamforming Feedback Matrix subfield is sent back)		

Figure A.1 Compressed Beamforming Report Information [2].

Size of V ($N_r \times N_c$)	Number of angles (N_a)	The order of angles in the Compressed Beamforming Feedback Matrix subfield
2×1	2	ϕ_{11}, ψ_{21}
2×2	2	ϕ_{11}, ψ_{21}
3×1	4	$\phi_{11}, \phi_{21}, \psi_{21}, \psi_{31}$
3×2	6	$\phi_{11}, \phi_{21}, \psi_{21}, \psi_{31}, \phi_{22}, \psi_{32}$
3×3	6	$\phi_{11}, \phi_{21}, \psi_{21}, \psi_{31}, \phi_{22}, \psi_{32}$
4×1	6	$\phi_{11}, \phi_{21}, \phi_{31}, \psi_{21}, \psi_{31}, \psi_{41}$
4×2	10	$\phi_{11}, \phi_{21}, \phi_{31}, \psi_{21}, \psi_{31}, \psi_{41}, \phi_{22}, \phi_{32}, \psi_{32}, \psi_{42}$
4×3	12	$\phi_{11}, \phi_{21}, \phi_{31}, \psi_{21}, \psi_{31}, \psi_{41}, \phi_{22}, \phi_{32}, \psi_{32}, \psi_{42}, \phi_{33}, \psi_{43}$
4×4	12	$\phi_{11}, \phi_{21}, \phi_{31}, \psi_{21}, \psi_{31}, \psi_{41}, \phi_{22}, \phi_{32}, \psi_{32}, \psi_{42}, \phi_{33}, \psi_{43}$
5×1	8	$\phi_{11}, \phi_{21}, \phi_{31}, \phi_{41}, \psi_{21}, \psi_{31}, \psi_{41}, \psi_{51}$
5×2	14	$\phi_{11}, \phi_{21}, \phi_{31}, \phi_{41}, \psi_{21}, \psi_{31}, \psi_{41}, \psi_{51}, \phi_{22}, \phi_{32}, \phi_{42}, \psi_{32}, \psi_{42}, \psi_{52}$
5×3	18	$\phi_{11}, \phi_{21}, \phi_{31}, \phi_{41}, \psi_{21}, \psi_{31}, \psi_{41}, \psi_{51}, \phi_{22}, \phi_{32}, \phi_{42}, \psi_{32}, \psi_{42}, \psi_{52}, \phi_{33}, \phi_{43}, \psi_{43}, \psi_{53}$

Figure A.2 Order of angles in the Compressed Beamforming Feedback Matrix subfield[2].

Channel Width	N_g	N_s	Subcarriers for which Compressed Feedback Beamforming Matrix subfield is sent: $scidx(0), scidx(1), \dots, scidx(N_s-1)$
20 MHz	1	52	-28, -27, -26, -25, -24, -23, -22, -20, -19, -18, -17, -16, -15, -14, -13, -12, -11, -10, -9, -8, -6, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28 NOTE—Pilot subcarriers ($\pm 21, \pm 7$) and DC subcarrier (0) are skipped
	2	30	-28, -26, -24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -4, -2, -1, 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28
	4	16	-28, -24, -20, -16, -12, -8, -4, -1, 1, 4, 8, 12, 16, 20, 24, 28
40 MHz	1	108	-58, -57, -56, -55, -54, -52, -51, -50, -49, -48, -47, -46, -45, -44, -43, -42, -41, -40, -39, -38, -37, -36, -35, -34, -33, -32, -31, -30, -29, -28, -27, -26, -24, -23, -22, -21, -20, -19, -18, -17, -16, -15, -14, -13, -12, -10, -9, -8, -7, -6, -5, -4, -3, -2, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58 NOTE—Pilot subcarriers ($\pm 53, \pm 25, \pm 11$) and DC subcarriers (0, ± 1) are skipped.
	2	58	-58, -56, -54, -52, -50, -48, -46, -44, -42, -40, -38, -36, -34, -32, -30, -28, -26, -24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -4, -2, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58
	4	30	-58, -54, -50, -46, -42, -38, -34, -30, -26, -22, -18, -14, -10, -6, -2, 2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58

Figure A.3 Subcarriers for which a Compressed Beamforming Feedback Matrix subfield is sent back [2].

Average SNR of Space-Time Stream i subfield	$AvgSNR_i$
-128	≤ -10 dB
-127	-9.75 dB
-126	-9.5 dB
...	...
+126	53.5 dB
+127	≥ 53.75 dB

Figure A.4 Average SNR of Space-Time Stream subfield [2].

A.2 MU Exclusive Beamforming Report

This field is present in the Compressed Beamforming Action Frame only for the MU scenarios. The MU Exclusive Beamforming Report information consists of Delta SNR subfields for each space-time stream (1 to N_c) of a subset of the subcarriers typically spaced $2N_g$ apart, where N_g is signaled in the Grouping subfield of the VHT MIMO Control field, starting from the lowest frequency subcarrier and continuing to the highest frequency subcarrier [2]. These Delta SNR subfields are ordered within the Multi User Exclusive Beamforming Report field as shown in Figure A.5. This figure shows a part of the whole table defined in the 802.11ac standard [2]. The equation mentioned in the table A.5 is given as follows:

$$\Delta SNR_{k,i} = \min(\max(\text{round}(10\log_{10}\left(\frac{\|H_k V_{k,i}\|^2}{N}\right) - \overline{SNR}_i), -8), 7) \quad (\text{A.1})$$

where,

k is the subcarrier index in the range of $\text{sscidx}(0), \dots, \text{sscidx}(N'_s-1)$

i is the space-time stream index in the range of $1, \dots, N_c$

H_k is the estimated MIMO channel for subcarrier k

$V_{k,i}$ is column i of the beamforming matrix V for subcarrier k

N is the average noise plus interference power, measured at the beamformee, that was used to calculate $\overline{SNR}_i$

$\overline{SNR}_i$ is the average SNR of space time stream i reported in the VHT Compressed Report Beamforming Report information.

Field	Size (Bits)	Meaning
Delta SNR for space-time stream 1 for sub-carrier $k = sscidx(0)$	4	$\Delta SNR_{sscidx(0), 1}$ as defined in Equation (8-2)
...	...	...
Delta SNR for space-time stream N_c for subcarrier $k = sscidx(0)$	4	$\Delta SNR_{sscidx(0), N_c}$ as defined in Equation (8-2)
Delta SNR for space-time stream 1 for sub-carrier $k = sscidx(1)$	4	$\Delta SNR_{sscidx(1), 1}$ as defined in Equation (8-2)
...	...	...

Figure A.5 MU Exclusive Beamforming Report information[2].