Modelling and Evaluation of 60 GHz IEEE 802.11 Wireless Local Area Networks in *ns-3*

A thesis
submitted in fulfillment
of the requirements for the degree
of

Master of Science
in
Computer Science

at
The University of Waikato

by
Michael May

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Department of Computer Science
Hamilton, New Zealand
March 5, 2014

Abstract

In this thesis we present modifications made to the popular network simulation environment *ns-3* to provide accurate simulation of IEEE 802.11ad Wireless Local Area Networks (WLANs) in the 60 GHz band. There is a need for such a framework as it allows research into how a directional, high performance wireless link affects various parts of the networking stack and Medium Access Control (MAC) design.

The work contained herein describes changes made to the existing WLAN MAC and Physical Layer (PHY) model in *ns-3* to support antenna directionality and multi-Gbps throughput. The resulting model is then analysed and found to accurately match optimal theoretical values in a number of test scenarios.

The result of this work is a simulation model capable of emulating IEEE 802.11ad WLANs with correct MAC and PHY representations.

Acknowledgements

My greatest thanks go to Dr. Dean Armstrong for his valuable support and guidance to carry out this research endeavour, and giving me the opportunity to develop and work with him. I sincerely appreciate the help Dr. Richard Nelson has given along this journey.

I am grateful to my parents, John and Karen, who have encouraged me to achieve above and beyond. Finally, I am thankful for the love, patience and support of my dearest Sarah.

I am also thankful for the efforts made by Andreas Löf, Brad Cowie and Mathew Button. They have been an invaluable source of help in this work.

This work has been supported financially by Virscient Ltd; Callaghan Innovation; The University of Waikato; and the Department of Computer Science, and I am indebted to these organisations.

Acronyms

A-BFT Association Beamforming Training

AC Access Category

AC_BE Best Effort Access Category

ACK Acknowledgement

AGC Automatic Gain Control

AIFS Arbitration Interframe Space

AIFSN Arbitration Interframe Space Number

A-MPDU Aggregate MAC Protocol Data Unit

A-MSDU Aggregate MAC Service Data Unit

AP Access Point

API Application Programming Interface

ARP Address Resolution Protocol

ATI Announcement Transmission Interval

AWV Antenna Weight Vector

BA Block Acknowledgement

BAR Block Acknowledgement Request

BC Beam Combining

BPSK Binary Phase Shift Keying

BRP Beam Refinement Protocol

BSS Basic Service Set

BTI Beacon Transmission Interval

CBAP Contention-Based Access Period

CCA Clear Channel Assessment

CE Channel Estimation

CPHY Control PHY

CRC Cyclic Redundancy Check

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTS Clear to Send

dB decibel

dBi decibel Isotropic

DCF Distributed Coordination Function

DIFS DCF Interframe Space

DMG Directional Multi-Gigabit

DBPSK Differential Binary Phase Shift Keying

DTI Data Transmission Interval

EDCA Enhanced Distributed Channel Access

FCS Frame Check Sequence

Gbps Gigabit per second

IBSS Independent Basic Service Set

IE Information Element

IEEE Institute of Electrical and Electronics Engineers

IP Internet Protocol

ISM Industrial, Scientific and Medical

ISS Initiator Sector Sweep

LLC Link Logical Control

MAC Medium Access Control

Mbps Megabit per second

MCS Modulation and Coding Scheme

MID Multiple sector ID Detection

MIMO Multiple In Multiple Out

MLME MAC Sublayer Management Entity

MPDU MAC Protocol Data Unit

MSDU MAC Service Data Unit

MTU Maximum Transmission Unit

NAV Network Allocation Vector

OFDM Orthogonal Frequency-Division Multiplexing

PHY Physical Layer

PLME PHY Sublayer Management Entity

PPDU PHY Protocol Data Unit

PSDU PHY Service Data Unit

PSK Phase Shift Keying

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

RSS Responder Sector Sweep

RTS Request to Send

RxSS Receive Sector Sweep

SC Single Carrier

SIFS Short Interframe Space

SLS Sector-Level Sweep

SME Station Management Entity

SNR Signal-to-Noise Ratio

SP Service Period

SQPSK Staggered Quadrature Phase Shift Keying

SSID Service Set Identifier

SSN Starting Sequence Number

SSW Sector Sweep

SSW-ACK Sector Sweep Acknowledgement

SSW-FB Sector Sweep Feedback

STA Station

STF Short Training Field

TBTT Target Beacon Transmission Time

TCP Transmission Control Protocol

TRN Training

TU Time Unit

TxOP Transmit Opportunity

TxSS Transmit Sector Sweep

UDP User Datagram Protocol

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

Contents

List of Figures

List of Tables

1

Introduction

The Institute of Electrical and Electronics Engineers 802.11-2012 WLAN standard [1] describes one of the most popular networking technologies in use today, with a large range of implementers and availability in most consumer computing devices currently sold. IEEE 802.11-based WLAN technologies have seen tremendous growth over nearly two decades of use due to its ease of operation, adequate data rates and ubiquity, security, and architecture that make it attractive to both commercial and residential users. WLANs are seen as an attractive tool in the home and workplace as they allow user freedom, especially as mobile computing devices become increasingly widespread.

There have been many amendments to the original IEEE 802.11 standard over the years in order for it to remain competitive and continue advancing alongside available technologies. The most notable amendments are those that directly modify the MAC and PHY for enhanced speed and operation, such as the 802.11n-2009 [2] and 802.11ac-2013 [3] amendments.

The 57-66 GHz band (henceforth referred to as the ”60 GHz band”) presents one of the largest license-free allocations in the radio spectrum coordinated around the world. The use of this band allows multi-Gbps wireless communication throughputs by using very large bandwidths. The 802.11ad amendment introduces operation of the 802.11 standard in this 60 GHz band. By using 2 GHz channel widths, multi-Gigabit per second (Gbps) throughputs are achievable with only a single spatial stream and relatively simple modulation schemes.

Due to atmospheric oxygen there is high attenuation in the 60 GHz band [4], and with transmit powers set by various country regulations typically around 10 mW a focus of the amendment has been allowing user-friendly operation in the face of these challenges. Antenna beamforming procedures are used to increase link margin to peers and have an added effect of allowing spatial reuse for denser deployments. Various applications for the 60 GHz band are envisioned; such as HDMI cable replacement, wireless docking, and high speed file synchronisation between devices.

While the 802.11ad amendment provides a framework for the beamforming procedures, a large amount of behaviour is undefined and left to the implementer. Network simulation is an attractive choice for rapid development of these undefined behaviours as it allows for fast and cost effective prototyping.

Network simulation allows the prediction of network behaviour without a real-world implementation. Various components that form a communications stack can be modelled, such as addressing and routing protocols, channels of communication, and the networking hardware that use these channels. This is an important research tool as it gives researchers the ability to quickly investigate how changing aspects of the networking stack affect operation under desired conditions [5], [6]. This thesis extends the network simulator *ns-3* to support 60 GHz WLANs based on the IEEE 802.11ad-2012 amendment [7].

ns-3 [8] is an open source network simulator primarily targeted at research use. Significant work has been done over the years to build *ns-3* into a very capable simulator. With its roots founded in *ns-2* [9], *ns-3* borrows a lot of the concepts and implementations of prior open source simulators and improves upon them in a number of different ways [10]. The IEEE 802.11 WLAN implementation in *ns-3* is based on a prototype project for simulating 802.11a networks called *yans* [11].

1.1 Related Work and Problem Foundation

During the standardisation process of the 802.11ad amendment, work was undertaken to model various characteristics of the 60 GHz channel based on experimental measurements [12]. Properties modelled include: space-time characteristics of the propagation channel; polarisation characteristics of antenna; time-dependent channel variations; and directional antenna properties. While this provides a very detailed investigation into questions regarding physical aspects and implementations there is no evaluation of the MAC design.

Other work has: looked at 60 GHz cell-based networks to provide whole-home coverage [13]; analysed 60 GHz link performance in various scenarios with differing physical characteristics versus antenna orientations [14]; and investigated the potential use of directional 802.11ad links to augment data centre networks through practical and simulated experiments [15].

While a lot of the physical aspects surrounding the 60 GHz band are understood there is a lack of real application of the MAC processes and knowledge of how effectively an 802.11ad device operates. There is a need for a simulation framework capable of testing the operation of a 802.11ad device and investigating how it copes with its responsibilities as it allows researchers and manufacturers to quickly prototype 60 GHz PHY and MAC designs.

ns-3 has shown good performance when compared to other open source simulators [6] and it has a well featured and proven 802.11 base to build upon. For these reasons it was chosen as the simulator platform for this work.

1.2 Thesis Contribution

In this thesis the existing *ns-3* architecture is extended to support simulation of networks comprising nodes based on the IEEE 802.11ad amendment. This work is broken down into three parts: an antenna model with variable beamwidth and movable boresight; implementation of a PHY model with mutli-Gbps transmission rates; and the design of various MAC-level components that allow standards-based operation in key areas, such as antenna beamforming procedures.

These aspects of the model are then validated and prove the model is capable of providing accurate results to aid in future research efforts. Analyses are undertaken using Transmission Control Protocol (TCP) and User

Datagram Protocol (UDP) to demonstrate that theoretically achievable and simulated throughputs correctly align, proving the validity and correct operation of various parts of the network stack. The beamforming procedures are shown to effectively find optimal antenna configurations for use in unicast communications.

The outcome of this work is a simulation environment that allows others to test and evaluate various MAC-level design parameters of a 802.11ad network device for research and development purposes. Beamforming procedures can be analysed and adjusted rapidly, antenna models can be investigated to see how they perform in conjunction with the MAC, and the effects of such high bandwidths on the upper networking layers can be examined.

1.3 Thesis Outline

The structure of the remainder of this thesis is as follows.

Chapter 2 provides an overview of how an IEEE 802.11 WLAN operates and details the changes brought forth by the IEEE 802.11ad-2012 amendment.

Chapter 3 introduces *ns-3* and its design. The focus of this chapter is explaining the existing WLAN module and the various components necessary for operation.

Chapter 4 details the modifications made to the WLAN simulation model in order to support the 802.11ad amendment. This information is divided into three sections: a steerable antenna model is presented for testing and evaluation purposes; modifications to PHY for supporting the new transmission rates are described; and the necessary MAC additions are explained.

Chapter 5 validates the model through tests designed to evaluate the modifications. By comparing expected theoretical results against output from the simulator we find that the modifications allow accurate simulation of 802.11ad networks.

Finally, Chapter 6 presents the conclusions of this work. and discusses areas where future work may be possible.

2

IEEE 802.11 WLANs

This chapter provides a broad overview of critical parts of the 802.11 standard, and the important changes made by the 802.11ad amendment in order to support multi-Gbps throughputs in the 60 GHz band. Section 2.1 explains the core concepts and functions of the base Medium Access Control (MAC) and Physical Layer (PHY), along with various existing methods used to increase throughput through efficiency. Section 2.2 outlines some of the issues that arise when using the 60 GHz band and the modifications to the MAC and PHY necessary for operating with these issues.

2.1 IEEE 802.11

First published in 1997, the IEEE 802.11 standard originally specified 1 Mbps and 2 Mbps raw bit rates transmitted over either infrared or the 2.4 GHz Industrial, Scientific and Medical (ISM) band using direct-sequence spread spectrum and frequency-hopping spread spectrum radios. Over time, amendments have added a second frequency range in parts of the 5 GHz band, increased raw bit rates [16] [17] [18] [2], and introduced various MAC mechanisms that aim to improve throughput [19] [2].

The most recent amendment to the standard, IEEE 802.11ac-2013 [3], introduces many MAC and PHY changes designed to push theoretical throughput in the 5 GHz band up to almost 900 Mbps in a single stream of data, and can support almost 7 Gigabit per second (Gbps) with multiple data streams.

Figure 2.1 gives a conceptual overview of a network stack. Packets typically start and end at the application. For example, a device requests a web page off a remote server across a network. The request progressively moves through the network stack, building a packet that encapsulates this request as it goes, which is then transmitted through the network by the physical layer. The 802.11 standard defines the physical layer and a majority of the link layer with the PHY and MAC specifications, respectively. 802.11 specifies differing ways for devices to physically access the medium through various types of PHY. A MAC layer logically sits above the PHY and dictates when and how a packet is sent to the PHY. All Wireless Local Area Network (WLAN) devices on the same logical network use a shared communication channel that may propagate a transmission from only a single transmitter.

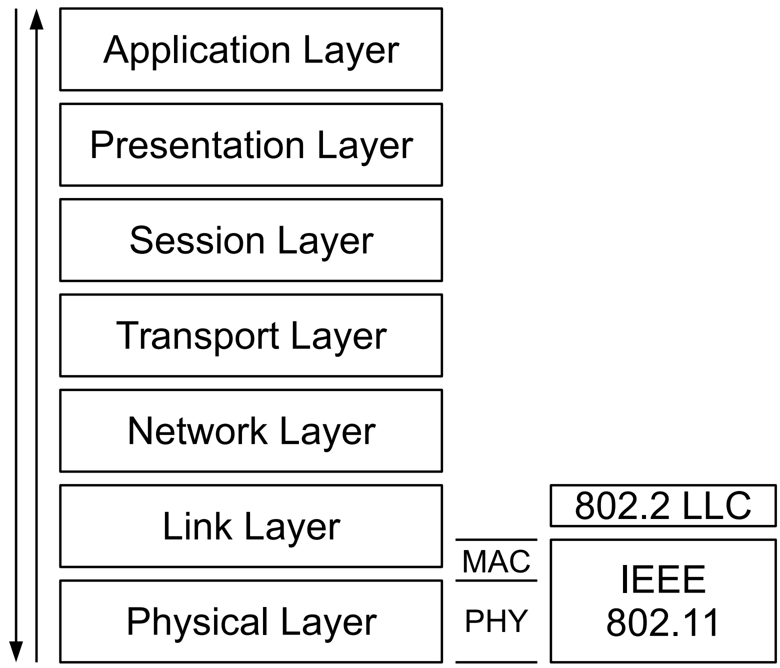

Figure 2.1: OSI model and location of IEEE 802.11 MAC and PHY.

2.1.1 Medium Access Control

The MAC protocol is a bridge between the upper networking layers and the PHY. The 802.11 MAC provides various functions for fair channel access and link management. As shown in Figure 2.2 the MAC is comprised of various parts that work together to send and receive data packets.

WLAN devices are generally half-duplex when transmitting and receiving

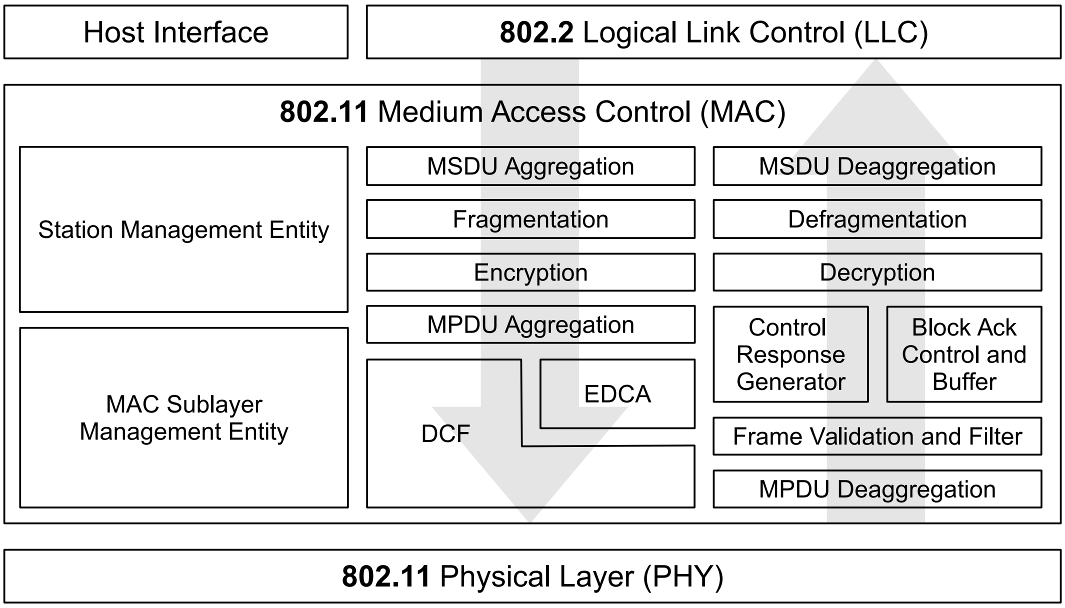

Figure 2.2: Overview of the internal 802.11 MAC structure

on the same channel. If multiple devices were to access the same channel at the same time then it could lead to interference where both packets may not be received correctly and the receivers drop them. The MAC tries to avoid these situations by having a device contend for medium access. When the channel becomes idle the MAC listens for a period of time to see if any other Stations (STAs) begin to use it. If the channel is idle then the device may begin transmission. Collisions may still occur if multiple devices begin to transmit at the same instant, in which case the link layer acknowledgement mechanism will case a retry.

On the transmitter side the core of the MAC is the Distributed Coordination Function (DCF) which implements Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to fairly share access to the channel without explicit coordination between devices. If a STA is wanting to transmit a packet it first performs a Clear Channel Assessment by listening to the channel for the DCF Interframe Space (DIFS) duration. If the channel has remained idle for that period then it takes ownership and begins transmission. If the channel became busy while the STA was waiting then when the channel again becomes free the STA will defer DIFS duration plus a random backoff period. If the channel remained idle for that length of time the STA will begin transmission, otherwise the STA will again defer DIFS plus what is remaining of the original backoff period.

The STA generates the random backoff period by choosing a number at random between zero and the current contention window value. This corresponds to the number of slot durations to wait after DIFS. The contention window is the upper bound slot value and is increased with every unsuccessful transmission, and reset on a successful transmission. There are minimum and maximum contention window values that vary between PHY types. DIFS and the slot duration are PHY-dependent time values.

Enhanced Distributed Channel Access (EDCA) builds on DCF to provide differential Quality of Service (QoS). Packets can be assigned to an Access Category (AC) which provides different timing parameters to give weighted access.

802.11 frames carry a duration field that indicates that the sender has reserved the medium for a period of microseconds. The transmitter sets the duration to a value it expects will allow it to complete the current transaction. For example, if it is a data frame then the duration may be set to cover the time taken for the recipient to transmit an Acknowledgement (ACK) frame back including the interframe spacing. The ACK frame would then have a duration of zero as there is nothing left to transmit in that transaction. Listening stations count down from the duration to zero and once zero they should expect the medium to be idle and their contention process can begin. Any additional packets in the exchange will update the countdown. This is a virtual carrier sensing mechanism called the Network Allocation Vector (NAV) and is helpful when dealing with the hidden node problem where not all stations can hear each other [20]. Devices can also use Request to Send (RTS)/Clear to Send (CTS) sequences to ensure all those within range of both points understand the medium is reserved.

On reception of a packet the receiver may respond appropriately based on the packet type. ACK frames are used to tell the originator that the frame was correctly received without error. In general, unicast data frames require an immediate acknowledgement except when a block acknowledgement agreement is in place for that AC. Management type frames generally elicit an immediate acknowledgement and may further require a response depending on the frame subtype. Block acknowledgement allows selective acknowledgement of multiple frames at once, and requires varying amount of state to be kept to allow retransmissions and reordering of frames.

As illustrated in Figure 2.3 a packet that enters the MAC is called a MAC Service Data Unit (MSDU) until it has a 802.11 MAC header prepended, at which point it is then a MAC Protocol Data Unit (MPDU). The MPDU is what is sent to the PHY for transmission. To the PHY any packet that enters for transmission is called a PHY Service Data Unit (PSDU), but once the preamble, PHY header and any trailing training fields are added then it becomes a PHY Protocol Data Unit (PPDU) as is transmitted out of the antenna. These data unit names also apply to received frames.

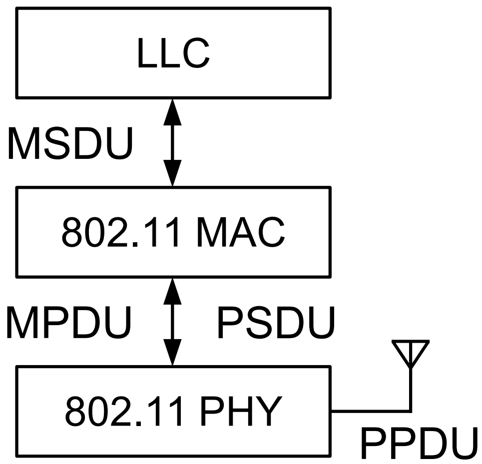

Figure 2.3: MAC and PHY data units and movement

Packet aggregation improves MAC efficiency by combining multiple packets into the same PHY transmission. Overall this can reduce the latency in transmitting a frame as it eliminates a lot of the interframe spaces used when transmitting an individual MSDU or MPDU.

In addition to the data plane functions outlined above, the 802.11 standard defines a management architecture comprised of three parts: the MAC Sublayer Management Entity (MLME); the PHY Sublayer Management Entity (PLME); and the Station Management Entity (SME). The MLME and PLME provide layer-specific management service interfaces through which management functions may be called.

The SME is the governing body that looks after the 802.11 network interface. It talks to an upper layer such as an operating system. The duties of the SME include coordinating channel scanning, power management, network association, and Basic Service Set (BSS) management.

Basic Service Set

A STA is any device that has the capability to use the 802.11 protocol, such as a computer, phone, or an access point. The BSS is a logical set comprised of all the connected STAs within a 802.11 network. Every BSS has a Service Set Identifier (SSID) associated with it for identification.

The most commonly used type of BSS is the infrastructure BSS involving an Access Point (AP) which STAs associate to. The AP acts as a bridge when STAs want to communicate in the network. There are two other types of BSS: the Independent Basic Service Set (IBSS); and the mesh BSS. These will not be discussed further in this thesis as they are little used and the mesh BSS is not supported in the 802.11ad amendment.

802.11 Framing

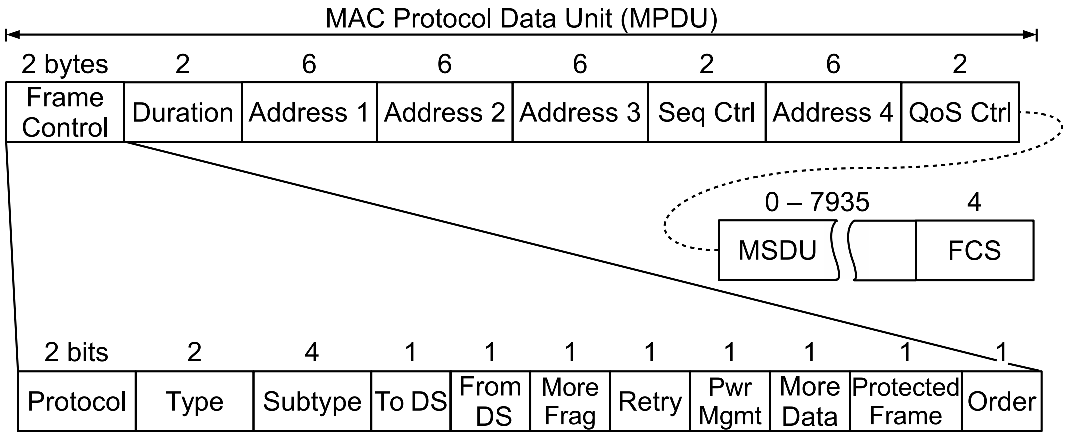

Figure 2.4: Structure of a 802.11 frame

The 802.11 MAC header is made up of various link layer-specific fields to account for addressing, various frame types, fragmentation and data reliability. Figure 2.4 shows the structure of a MPDU. The two octets of Frame Control contains descriptors of the MPDU and tell the receiver how it should interpret the MSDU. Four address fields allow for standard unicast, multicast and broadcast transmissions. These fields have various purposes depending on the frame type and in some cases only a subset are included. The Sequence Control field is used in frame identification. Up to 7935-octets of payload is allowed. The last four octets are a Cyclic Redundancy Check (CRC) of the MPDU called the Frame Check Sequence, which is used for frame validation.

Every 802.11 frame contains a PHY-specific header at the front transmitted at a very low bit rate in order for legacy devices to understand that the medium is being accessed. This contains fields such as the preamble used to train the receiver onto the signal, and information fields that define what speed the payload is modulated at and how to decode it.

Management

In an infrastructure BSS, Beacon frames are sent periodically by the AP (commonly every 100 Time Units (TUs) [21], where one TU is 1.024 milliseconds). The beacon period defines a series of Target Beacon Transmission Times (TBTTs) at which the AP will begin contention to transmit a Beacon frame. The role of the Beacon is to announce the existence of, and carry information about the BSS. Some of the information conveyed includes a timestamp to help stations synchronise their internal timers, capability information of the AP, and parameter sets for the supported physical layers. Figure 2.5 illustrates the transmission of Beacon frames at every TBTT.

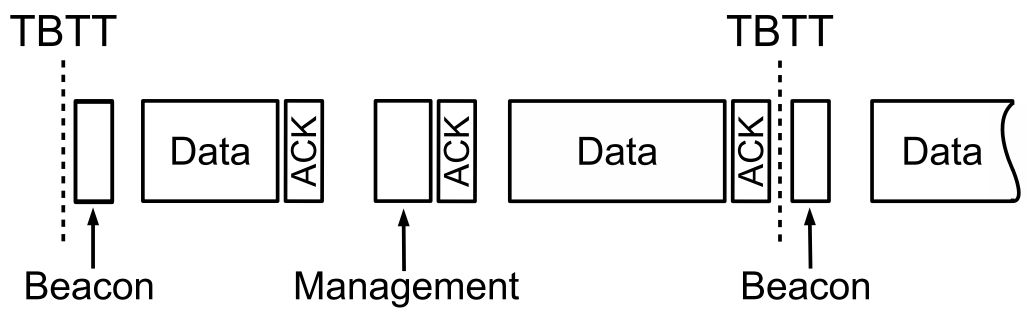

Figure 2.5: Beacons transmissions at the TBTT

Before a STA is able to transmit and receive data in a BSS it needs to associate with the AP. Following a generally null authentication exchange, the STA transmits an Association Request frame to the AP. If the STA is allowed to join the BSS then an Association Response frame is sent back. Using these frames the STA and AP exchange capabilities and the AP informs the STA of the BSS operating parameters.

In most 802.11 installations authentication is required before a STA can associate with the BSS. This only allows those STAs that are authorised to use the network. Security is typically based on the IEEE 802.1X standard [22].

Enhanced Distributed Channel Access

Enhanced Distributed Channel Access [19] is an extension of the DCF that provides prioritisation to frames requiring differentiated QoS. Four ACs are defined, each with different timing parameters relating to channel access that statistically prioritise access between ACs.

DIFS is the standard waiting period used when a STA is testing the idleness of the channel. The value is calculated in Equation 2.1, where Short Interframe Space (SIFS) and the slot time may vary between PHY types.

$$DIFS = SIFS + (2 \times slotTime) \tag{2.1}$$

$$AIFS = SIFS + (AIFSN \times slotTime) \tag{2.2}$$

As shown in Equation 2.2, EDCA introduces the Arbitration Interframe Space (AIFS) in place of DIFS, this uses the variable Arbitration Interframe Space Number (AIFSN) which varies between ACs. This leads to a variation in the minimum waiting period per AC. In addition, the minimum and maximum of the upper bound of the contention window has changed in favour of the higher priority ACs.

Another important feature is the Transmit Opportunity (TxOP) that allows contention-free access for a period of time. A STA contends for access as normal and, depending on the AC, is then allowed to reserve the medium for a period up to a maximum time defined for that AC, using the duration field of the packets it sends.

Block Acknowledgements

In general, unicast data frames require an immediate acknowledgement. For large sequences of data transmissions there is significant overhead in sending an ACK response for each individual MPDU received. Also, for high data rates the overhead of contention and interframe spacings can easily dwarf the duration of the actual data transmission. This issue can be throughput limiting so a method of reducing MAC overheads has been standardised with block acknowledgements.

Block acknowledgements allow for multiple MPDUs to be acknowledged with a single frame. A block acknowledgement agreement is made between a pair of STAs that, for a particular traffic class, Block Acknowledgement

(BA) frames will be used in place of regular ACK frames. The are four types of the block acknowledgements: immediate; delayed; HT-immediate; and HT-delayed.

Under immediate block acknowledgement the originator of the data being acknowledged would send a Block Acknowledgement Request (BAR) that contained the Starting Sequence Number (SSN) that the recipient would use as the first frame to acknowledge. The recipient then immediately transmits a BA frame that acknowledges a number of frames after the SSN. The buffer size negotiated when the agreement was set up determines the number of frames sent by the originator before it sends a BAR. The BA frame contains a 1024-bit scoreboard that allows acknowledgement of up to 64 MSDUs, which can each be fragmented up to 16 times. For every successfully received MSDU the responder sets the appropriate bit in the scoreboard to indicate that sequence number is being acknowledged. Delayed BAs are identical in concept but operate in a delayed fashion. Instead of immediately responding to a BAR with a BA frame the BAR is normally acknowledged and on a subsequent channel access the BA frame is sent which is also normally acknowledged.

The 802.11n amendment introduced modified versions of these two mechanisms that further improved efficiency in light of other changes: HT-immediate block acknowledgement; and HT-delayed block acknowledgement. HT-delayed block acknowledgement allows the receiver to forgo responding to a BAR or BA with a normal ACK frame.

The HT-immediate protocol changes the behaviour of some key elements of the immediate block acknowledgement policy. A compressed version of the scoreboard has been added, eliminating the acknowledgement of fragments, thereby reducing it to only 64-bits. The purpose of the Block Ack Policy field in the QoS Control element when used in an aggregate has changed so that instead of differentiating between immediate and delayed BA frame, when an aggregate MPDU is transmitted it discerns between normal and HT-immediate BA frames. When set to normal ACK an implicit BA is sent by the responder. This behaviour further reduces overhead by eliminating the need for a BAR frame. The BA frame allows selective acknowledgement of the MPDUs received. This means that if interference or error was to occur when receiving part of the transmission if any of the individual MPDUs were received error free then they could be acknowledged.

Aggregation

Aggregation allows multiple MSDUs and MPDUs to be sent as a single transmission, reducing overhead caused by contention and interframe spacing, and thus increasing achievable throughput. The 802.11n amendment introduced two types of data aggregation: MSDU aggregation; and MPDU aggregation.

MSDU aggregation logically resides at the top of MAC and combines one or more MSDUs into an Aggregate MAC Service Data Unit (A-MSDU), which then becomes an MPDU with the addition of a MAC header. MSDUs of the same traffic class are joined together until the aggregate reaches a maximum allowed size. Each MSDU has a 14-octet subframe header prepended which provides the destination and source MAC address, along with a length field. Padding is appended to the MSDU to round it to a 32-bit word boundary.

MPDU aggregation logically occurs at the bottom of the MAC before the frame is queued for transmission. Multiple MSDUs are each given a 802.11 MAC header with a unique sequence number and these MPDUs are then concatenated into an Aggregate MAC Protocol Data Unit (A-MPDU). Each MPDU is prepended with a 4-octet delimiter which contains the length of the MPDU, an 8-bit CRC field for header corruption detection, and a constant signature (ASCII ”N”) to aid in delineation.

2.1.2 Physical Layer

The 802.11 standard defines several physical implementations using various technologies in the 2.4 and 5 GHz bands. Each PHY type is designed to share the same MAC as shown in Figure 2.6.

The spectral bands are divided into fixed sized channels to allow medium sharing between networks and to try and minimise interference. The 2.4 GHz band can be made of up to fourteen 20 MHz channels, depending on country regulations. Of these fourteen, only four channels are of sufficient distance to not overlap. The 5 GHz band has a lot more variance of unlicensed spectrum due to country regulations, but there is more bandwidth

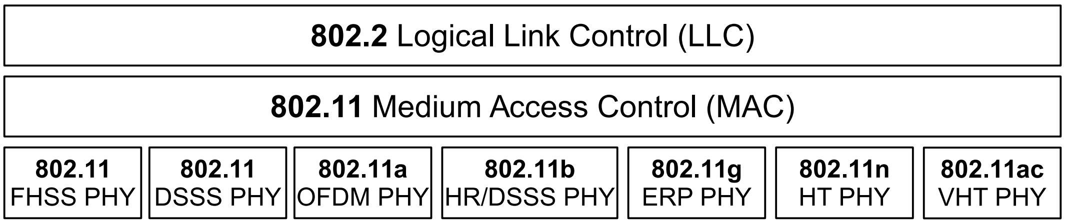

Figure 2.6: Overview of the 802.11 PHY rates

available and the band is currently less congested.

There are many different methods of encoding a data stream on to a waveform that each have their own particular advantages and disadvantages. 802.11 is fundamentally based on two different techniques: spread spectrum, and Orthogonal Frequency-Division Multiplexing (OFDM).

Spread spectrum techniques work by mathematically diffusing a narrow-band signal across a wider band of frequencies. This makes the signal appear more like noise and it becomes much more resilient to interference. The receiver reverses the operation, regenerating the original narrow-band signal and in the process cancelling most narrow-band noise that may have been added to the frame. Phase Shift Keying (PSK) methods are used to convey the data stream bits into changes in phase of a signal. Binary Phase Shift Keying (BPSK), or 2-PSK, is the simplest form of PSK which uses two phases separated by 180° to represent the binary bits but is only capable of two states or symbols, and therefore only 1-bit/symbol. Quadrature Phase Shift Keying (QPSK) uses four phases separated by 90° to provide 2-bits/symbol. These PSK methods are the basis of legacy 802.11 and provide low data rates due to a low spectral efficiency.

802.11a introduced a markedly different PHY implementation with the use of unlicensed 5 GHz spectrum and OFDM with Quadrature Amplitude Modulation (QAM), pushing bit rates up to 54 Mbps. 802.11g subsequently introduced use of these techniques in the 2.4 GHz band. OFDM works by dividing a channel into smaller subcarriers and splits the signal across each subcarrier in parallel. QAM builds upon PSK where data can not only be represented by phase angle but also amplitude. This leads to denser constellations and greater spectral efficiency. 802.11a allows BPSK, QPSK, 16-QAM and 64-QAM to be used in each subcarrier. 16-QAM means that there are 16 constellation points, each representing 4-bits. The 64-QAM constellation contains 64 points and 6-bits per point.

802.11n introduced a multitude of changes to the physical layer with the aim of providing higher throughput. Adjacent channels can be combined to be 40 MHz wide, resulting in data throughputs greater than that obtainable with 20 MHz. Spatial multiplexing was also introduced in the form of MIMO techniques that use an array of antennas, and more intelligent radio transceivers. This allows multiple streams of data to be transmitted simultaneously as the data is split across the streams, increasing throughput.

With multiple antennas it is possible to perform transmit beamforming where the transmitter applies digital pre-coding to the data streams to focus the antenna boresight in the direction of the receiver.

802.11ac builds on a lot of the techniques introduced with 802.11n to enable multi-Gbps throughputs. These include: increasing channel widths up to 160 MHz; allowing up to eight spatial streams; adding Multiple User Multiple In Multiple Out (MIMO) where space division multiple access is used to simultaneously service multiple STAs; and allowing the use of 256-QAM for very dense spectral encoding.

2.2 IEEE 802.11ad

The IEEE 802.11ad amendment builds upon the foundations of the 802.11 standard and enhances operation to allow multi-gigabit per second throughput in the 60 GHz band. This is accomplished through various MAC and PHY modifications and additions. The terms 802.11ad and Directional Multi-Gigabit (DMG) are used interchangeably to the refer to the IEEE 802.11ad amendment.

There have been two previously completed standards for operation at 60 GHz: IEEE 802.15.3c [23]; and ECMA-387 [24]. Both of these have been developed with the intention of very high throughput, short range communications in similar environments. 802.11ad differentiates itself from these two standards with two key design choices: fast session transfer between 802.11 PHY; and maintaining the 802.11 ecosystem, basic operation and interoperability. Fast session transfer permits multi-band devices to switch between operating bands and PHY while continuing the existing 802.11 session.

This section presents the 802.11ad Physical Layer first as it explains important concepts necessary for understanding some of the MAC design choices.

2.2.1 Physical Layer

Four channels are allocated centred at 58.32 GHz, 60.48 GHz, 62.64 GHz, and 64.8 GHz with a 1.88 GHz 0 dBr (relative to the maximum spectral density of the signal) bandwidth and 2.16 GHz spacings. The trade-off with using the 60 GHz band is the very high free space attenuation (5-30 dB/km) due to oxygen in the atmosphere [25] [26] [4], and even greater loss through materials [27].

An effect of the very short wavelength is that transmissions are much more directional than that of the sub-6 GHz bands and must be treated differently in order to maximise range and throughput. The MAC has allowances for transmit and receive beamforming in order to electronically optimise the antenna gain for a certain spatial direction, allowing devices to optimise signal paths for the greatest quality signal. In this band waves are more likely to be reflected and have a lower diffusion rate that those of lower wavelengths [28]. This property makes beamforming around objects possible in a non-line-of-sight environment.

Modulation Methods

As shown in Table 2.1, there are 32 different PHY rates based on four different types of modulation scheme. Each Modulation and Coding Scheme (MCS) vary in data rate, complexity and hardware requirement. Five mandatory MCSs are defined, MCS0-4, ranging in raw bit rate from 27.5 to 1155 Mbps.

The mandatory Control PHY (CPHY) (MCS0) is the most basic of the schemes and it is primarily used for transmitting management and control frames that require the greatest chance of reception. The CPHY uses Differential Binary Phase Shift Keying (DBPSK) with a coding rate of 1/2, giving a raw bit rate of to 27.5 Mbps. While the CPHY operates at a comparatively slow speed, these various modulation aspects help ensure reliable communication even with a low Signal-to-Noise Ratio (SNR).

There are four mandatory Single Carrier (SC) PHY (MCS1-4) based on $\pi/2$-

MCS	Modulation	Bit Rate (Mbps)
Control PHY (CPHY)		
0	DBPSK	27.5
Single Carrier (SC)		
1	$\pi/2$-BPSK	385
2	$\pi/2$-BPSK	770
3	$\pi/2$-BPSK	962.5
4	$\pi/2$-BPSK	1155
5	$\pi/2$-BPSK	1251.25
6	$\pi/2$-QPSK	1540
7	$\pi/2$-QPSK	1925
8	$\pi/2$-QPSK	2310
9	$\pi/2$-QPSK	2502.5
10	$\pi/2$-16QAM	3080
11	$\pi/2$-16QAM	3850
12	$\pi/2$-16QAM	4620
OFDM		
13	SQPSK	693
14	SQPSK	866.25
15	QPSK	1386
16	QPSK	1732.5
17	QPSK	2079
18	16-QAM	2772
19	16-QAM	3465
20	16-QAM	4158
21	16-QAM	4504.5
22	64-QAM	5197.5
23	64-QAM	6237
24	64-QAM	6756.75
Low Power SC		
25	$\pi/2$-BPSK	626
26	$\pi/2$-BPSK	834
27	$\pi/2$-BPSK	1112
28	$\pi/2$-QPSK	1251
29	$\pi/2$-QPSK	1668
30	$\pi/2$-QPSK	2224
31	$\pi/2$-QPSK	2503

Table 2.1: IEEE 802.11ad MCSs with modulation type and bit rate

BPSK modulation with code rates at 1/2, 5/8, and 3/4, provides bit rates from 385 Mbps to 1155 Mbps. There are eight additional optional SC PHY (MCS5-12) using $\pi/2$-BPSK, $\pi/2$-QPSK and $\pi/2$-16QAM modulation, and code rates ranging from 1/2 up to 13/16. These provide bit rates ranging from 1251.25 Mbps (MCS5) to 4620 Mbps (MCS12).

The OFDM PHY rates (MCS13-24) provide bit rates from 693 Mbps to 6756.75 Mbps using Staggered Quadrature Phase Shift Keying (SQPSK), QPSK, 16-QAM, and 64-QAM modulations, and code rates of 1/2 to 13/16.

An optional low power SC PHY (MCS25-31) provides lower powered processing requirements for DMG transceivers with mobile devices in mind while granting raw bit rates from 626 Mbps (MCS25) to 2503 Mbps (MCS31).

60 GHz Antennas

Designing antennas at 60 GHz presents challenges due to the short wavelength, and it is largely through advances in high-speed CMOS technology since the millennium that this band has become more accessible [29]. Single antenna solutions are typically unsuitable for use as they have issues with gain and directivity, making directional antenna solutions more desirable in highly mobile devices.

Antenna dimensions are inversely proportional to the carrier frequency [30], thus for a given area the antenna density can be a lot higher for the millimetre band than what is achievable with 2.4 and 5 GHz WLAN counterparts. This opens the door to technologies such as on-chip antenna structures that reduce footprints and decrease interconnect losses [31] [32].

Switched beam and phased array antenna systems are two popular ways of encompassing multiple antennas in order to help provide spatial coverage.

Switched beam antenna systems multiplex multiple antenna elements to the same RF front-end and the antenna in use is chosen through control signals. Complex antenna arrangements are possible through various levels of multiplexing but signal quality and loss due to construction and design must be taken into account. A scenario for this is that each antenna services a different spatial area so full coverage can be achieved through many antenna elements.

Phased arrays are another approach that uses phase shifting of the RF signal to force constructive and destructive interference in order to tune the radiation pattern in a particular direction. The array is made of multiple antenna elements that each have a controllable phase shifter inline. This allows direction, beamwidth and consequently gain [33] to all be adjusted.

2.2.2 Medium Access Control

Due to the 802.11ad amendments focus on different usage scenarios there has been a shift away from the WLAN idea of servicing a large area such as a house to more of a Wireless Personal Area Network (WPAN) focus. While the core functionality of the 802.11 MAC has been maintained there are significant additions and changes made in order to facilitate the different characteristics caused by the use of the 60 GHz band.

Figure 2.7 shows the key components in a 802.11ad MAC. All contention-based access occurs in the Contention-Based Access Period (CBAP) which solely uses EDCA queues. There is also the addition of a contention-free access method called a Service Period (SP). There a three defined periods at start of each beacon interval: the Beacon Transmission Interval (BTI); the Association Beamforming Training (A-BFT); and the Announcement Transmission Interval (ATI). These three periods are separate from the typical data flow and have direct access to the PHY while maintaining CSMA/CA principles.

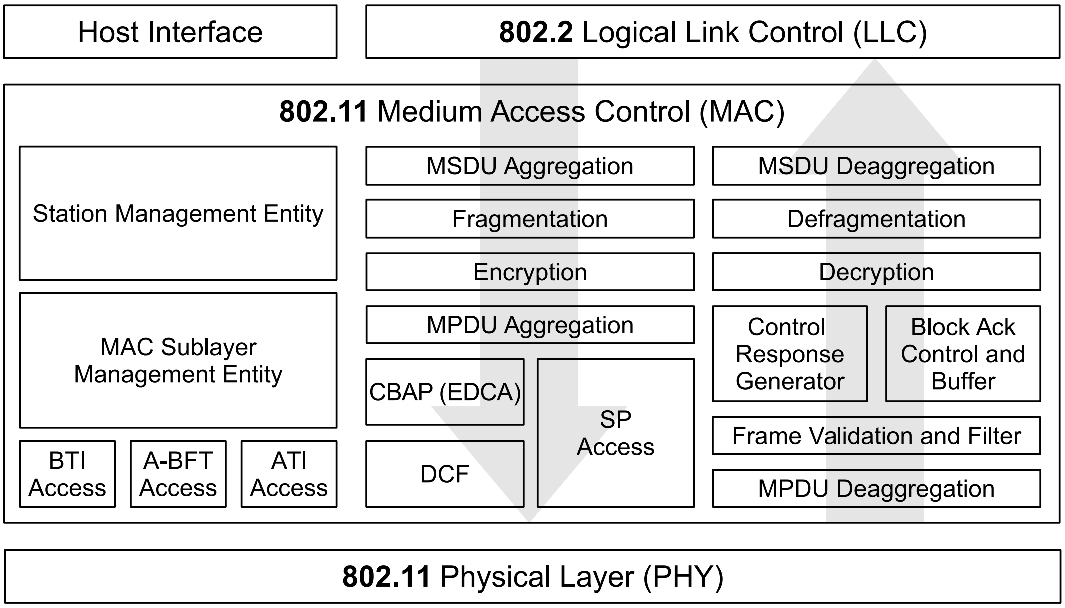

Figure 2.7: Overview of the internal 802.11ad MAC structure

Beamforming

Beamforming procedures are used by the MAC to compensate for the characteristics of the 60 GHz band and increase link margins between devices. There are three types beamforming procedures defined in the 802.11ad amendment: the Sector-Level Sweep (SLS); the Beam Refinement Protocol (BRP) phase; and Beam Tracking. The SLS is used for rough training and establishes a link suited for lower throughput transmissions. The BRP phase refines upon the rough training and can end in a highly optimised link. Beam Tracking can be used to monitor and adjust the trained link while transmitting non-beamforming frames. The 802.11ad MAC specifies three important concepts with regard to the antenna system from its perspective: a sector; an Antenna Weight Vector (AWV); and an antenna.

An antenna is a simple integer representation that allows the MAC to be ignorant to the antennas physical implementation. This means an antenna with multiple elements will logically appears as a single antenna to the MAC. The PHY would be aware of this abstraction and would know how to interpret it. The MAC is allowed up to four addressable antenna but the implementation could consist of many times that.

An AWV is a representation of an antenna configuration, such as amplitude and phase. The amendment defines little about an AWV so the MAC and PHY interpretations are largely implementation dependent. This concept is used exclusively in the BRP phase and Beam Tracking. A sector is a simple integer representation of an AWV that allows the MAC to perform basic beamforming procedures in the SLS. Each antenna is allowed up to 64 sectors and a combined total of 128 sectors across all four antenna.

During a SLS, each Sector Sweep (SSW) frame contains the antenna and sector used when transmitting the frame so that the feedback can inform the originator of the best antenna and sector used in the procedure. In the BRP and Beam Tracking procedures the transmit AWV is not conveyed. Instead basic feedback has to be communicated by an index into the transmission fields used for training. It is because of this that the amendment provides more concrete specifications for the antenna and sector concepts. All beamforming procedures are between two devices, one that initiates the procedure and the other that responds, appropriately termed the initiator and responder respectively.

Sector Level Sweep The Sector-Level Sweep is the initial stage of beamforming training, with the goal of establishing communications at the CPHY rate. The parameters found act as the starting point for beam refinement to improve upon. The SLS is made up of two parts: the Initiator Sector Sweep (ISS); and the Responder Sector Sweep (RSS). There are two types of SLS: the Transmit Sector Sweep (TxSS); and the Receive Sector Sweep (RxSS). The ISS and RSS can be either of these types, as described in Table 2.2.

	Initiator	Responder
Initiator TxSS	Tx from multiple sectors	Rx in best sector
Initiator RxSS	Tx from best sector	Receives in multiple sectors
Responder TxSS	Rx in best sector	Tx from multiple sectors
Responder RxSS	Receives in multiple sectors	Tx from best sector

Table 2.2: Sector-Level Sweep variations

The purpose of the TxSS is for the transmitting STA to learn its best transmit sector to the receiver. Multiple SSW frames are transmitted using a range of sectors. Each frame contains an identifier for the corresponding sector and antenna in order for the receiver to report back the optimal sector.

The RxSS is used by the receiver to find the best sector to receive from the peer. The transmitter sends the requested number of SSW frames from a either a quasi-omni directional configuration or the best transmit sector if known, while the receiver is changing its receive sector configuration between every frame.

Figure 2.8 illustrates the structure of the SLS. While the initiator is transmitting its chosen number of SSW frames the responder is either in a quasi-omni receive pattern or switching through its receive sectors. The reverse occurs during the RSS. At the conclusion of both the ISS and RSS feedback is sent from the SSW frame receiver to the transmitter. During the RSS each SSW frame contains a feedback Information Element (IE) for the initiator. Once the RSS has ended then the initiator transmits a Sector Sweep Feedback to the responder. If the SLS occurred outside of the A-BFT then a Sector Sweep Acknowledgement (SSW-ACK) frame is sent by the responder. The Sector Sweep Feedback (SSW-FB) and SSW-ACK frames may contain information requesting further beam refinement with the BRP phase.

Beam Refinement Protocol The goal of the Beam Refinement Protocol phase is to further improve the transmit and receive antenna configurations us-

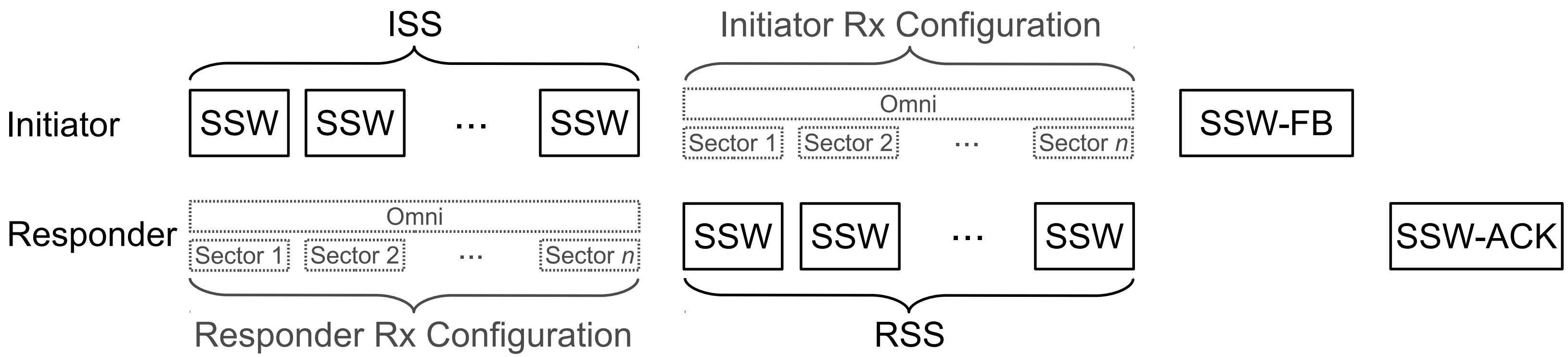

Figure 2.8: Structure of a Sector-Level Sweep

ing an iterative approach. The BRP phase consists of four subphases: BRP setup; Multiple sector ID Detection; Beam Combining; and BRP frame transactions.

The BRP setup subphase is used to exchange capability information and request either Multiple sector ID Detection (MID) or Beam Combining (BC) execution, or a combination. This subphase may be skipped if the BRP phase follows a SSW-ACK and no MID or BC was requested by either STA.

The MID and BC subphases are used to mitigate effects caused by imperfect quasi-omni antenna receive radiation patterns used in the SLS that may cause the incorrect choice of best transmit sector. MID involves a quasi-omni transmit pattern transmitting BRP frames to a receiver that is testing various receive AWVs. In BC the pair avoid using quasi-omni configurations and instead a set of receive and transmit AWVs are tested against each other.

BRP transactions involve exchanges of BRP frames with Training (TRN) fields appended. A TRN field is simply a fixed duration complementary sequence that allows the receiver to perform channel measurements to evaluate antenna configurations. The purpose is for the transmitter to either vary the AWV used to send each TRN field for transmit training, or maintain a constant transmit AWV if the receiver is performing receive training by changing its configuration every TRN field. The initiator and responder take turns transmitting a BRP frame with appended TRN fields until both sides have completed training. If transmit training was performed the feedback is part of the subsequent BRP response.

Beam Tracking Beam Tracking allows a device to continually assess its transmit and receive antenna configuration. It is a continuation of beamform training that involves TRN fields being appended to regular packets. Both receive and transmit training are allowed but instead of signalling the presence of TRN fields with BRP frames the PHY header is used. Transmit training still requires feedback so the next frame back to the originator has a BRP frame aggregated to it with the measurements.

Beacon Interval

An 802.11 infrastructure BSS has Beacon frames transmitted by the AP periodically associated STAs to synchronise. As outlined in Figure 2.9, 802.11ad features a more structured beacon interval where every interval includes allocated periods for transmission of multiple DMG Beacon frames (BTI), coarse beamforming procedures (A-BFT), management exchanges (ATI), and data transmissions (Data Transmission Interval (DTI)).

The Beacon Transmission Interval is a period where the AP sends one or more DMG Beacon frames used for synchronisation and initial beamforming training, as shown in Figure 2.10. The BTI is an initiator TxSS that uses DMG Beacon frames. Each frame is sent from a sector on an antenna and STAs can report feedback to the AP during the A-BFT.

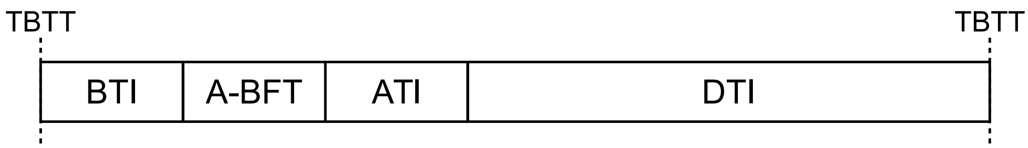

Figure 2.9: Structure of the beacon interval in IEEE 802.11ad

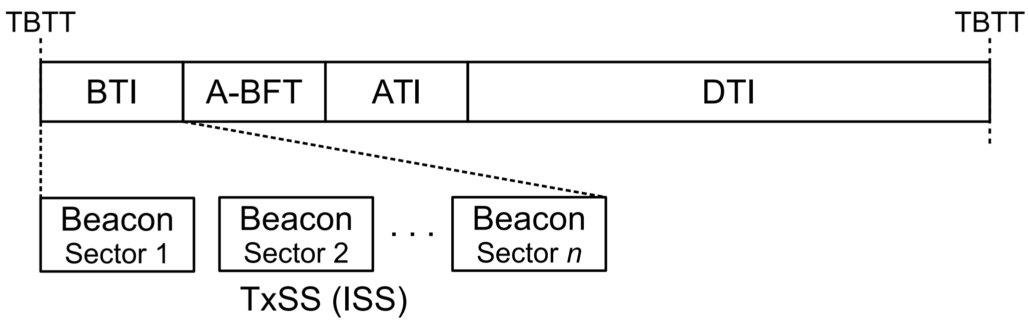

Figure 2.10: Structure of the BTI

The Association Beamforming Training period provides non-AP STAs the ability to perform beamforming training with the AP. The allocated time is divided into slots of fixed length and STAs contend for the opportunity to perform a sector sweep with the AP. Figure 2.11 shows the structure of the A-BFT including a RSS and the feedback from the AP.

The Announcement Transmission Interval is used by the AP to query a set of the associated STAs in case they need to talk to the AP. Some envisioned use cases for this interval are authentication, and SP requests. The request and response exchanges can take varying form depending on what is requested at either end, as shown in Figure 2.12.

The Data Transmission Interval is the allocated time for data exchanges. It is made up of Contention-Based Access Periods and allocated Service

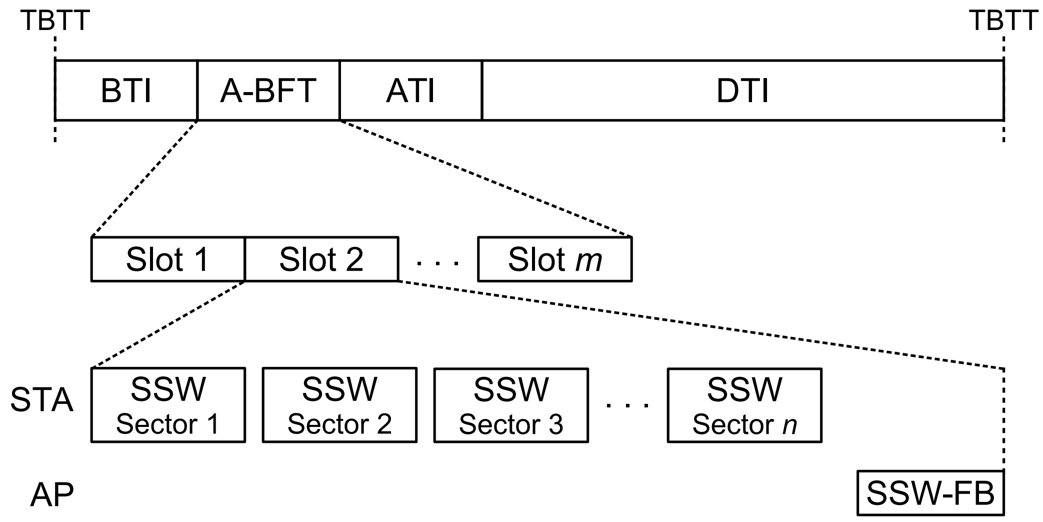

Figure 2.11: Structure of the A-BFT period

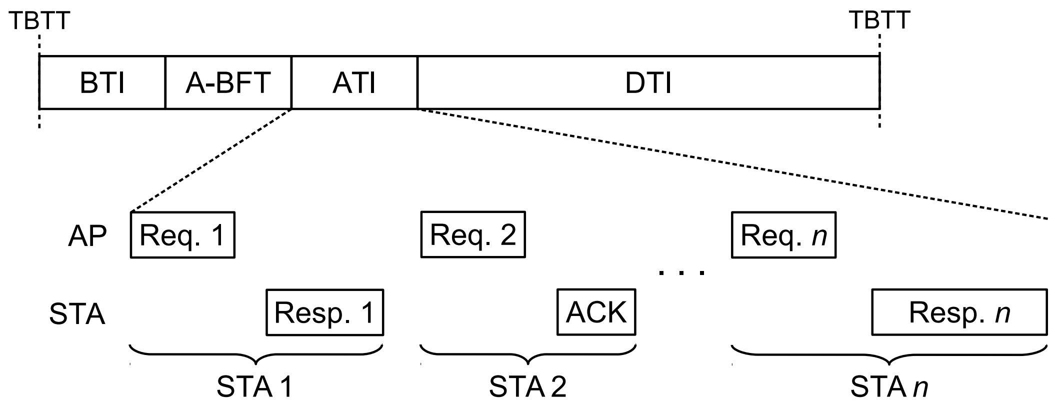

Figure 2.12: Structure of the ATI

Periods, by default all access is based on CBAPs. A CBAP is based upon the conventional CSMA/CA principle with DCF but uses EDCA for priorities and TxOPs. A SP is a contention free period between an initiator and responder. The initiator requests a SP with the destination and the AP may granted this depending on other events occuring in the DTI. A SP can be between any type of STA, AP or otherwise. As shown in Figure 2.13 the contents of a CBAP or SP can be the same, the difference is in the access method.

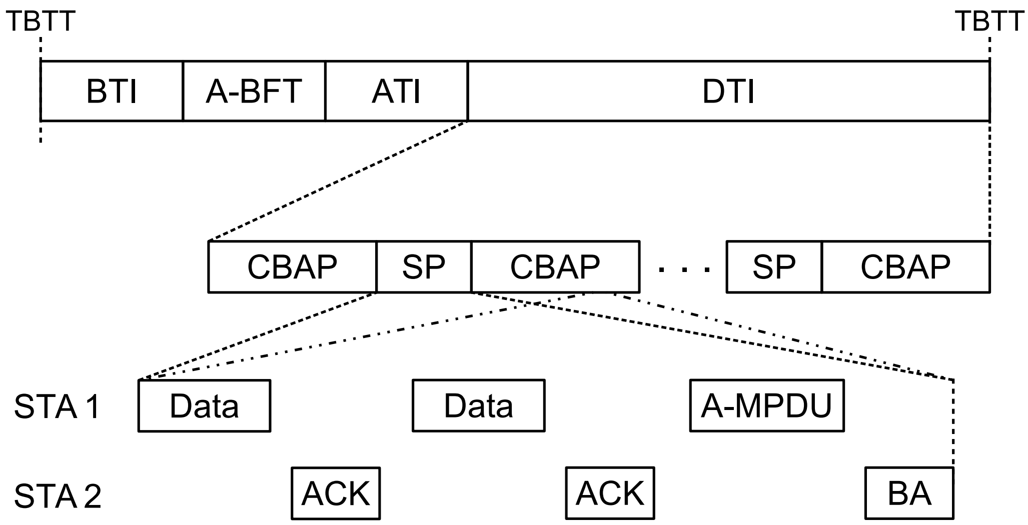

Figure 2.13: Structure of the DTI

3

ns-3 Network Simulator

ns-3 [8] is an open source, discrete event network simulator used to model and test network protocols and medium access control mechanisms. It boasts a large, active community, and encourages peer reviewed simulation models. *ns-3* was chosen for this work as has shown good performance when compared to other open source network simulators [6] and it has a well featured and proven 802.11 base to build upon.

This chapter explains how the *ns-3* framework operates with particular focus on the WLAN module used for IEEE 802.11 MAC and PHY simulations. Section 3.1 gives an overview of the abstract grouping of objects in *ns-3* and how they relate to each other. Section 3.2 details the basics of a network model and shows how an Ethernet network is created and operates. Section 3.3 gives a detailed explanation of the *ns-3* WLAN model and how the individual components interact.

The *ns-3* software infrastructure is designed to allow the creation of models that are realistic enough to be used in real-time simulation, with the option of integrating real-world networking stacks [34] [35]. Simulations can transmit and receive packets on real network interfaces, and can serve as an interconnection between virtual machines.

ns-3 is built as a C++ library and is linked to a scenario-specific main function which creates, initialises and begins the simulation. The software framework can be generalised as multi-layered with dependencies primarily going down the stack. As outlined in Figure 3.1 the user provided main function sits at the top and calls on objects below it; typically through helper functions that then call a multitude of lower level Application Programming Interface (API) functions. These helper functions provide an easy interface to object creation and initialisation. Their most common use is to interface and set up nodes and the necessary support objects, such as the simulated hardware, mediums and protocols, among others. The simulator handles all occurring events and provides timing and triggering. The simulation core defines components that are common across all models such, as smart pointers, logging and attributes.

Events are added to the scheduler through a number of various ways. A simulation may start and stop an application using specific time values of various granularity, while a part of the model may schedule an event for a particular time in the future based on prior events. When the globally accessible function Simulator::Run() is called by the main function the simulator iterates linearly through each scheduled event and performs the appropriate action. Many events trigger others such as the sending of a packet triggering the receipt of that packet in another node. This in turn, may trigger further events through processing and acknowledgement. Events are generally not performed in real-time; instead the simulated time is fast-forwarded to the next event, allowing runtime to be decreased. The simulation stops when the event queue becomes empty or when the simulation is explicitly terminated via a call to Simulator::Stop().

3.1 Common Object Types

All models in *ns-3* are written in C++ and the environment has been designed to allow maximum code reuse and ease of integration. A simulation is composed of objects of various types that perform specific actions, as detailed below.

Node A node is an object that can be thought of as a bare computer, which objects such as applications, protocol stacks and NetDevices can be added to. A group of nodes that are to have the same functionality may be grouped in a specific container, for instance the stations in a wireless network.

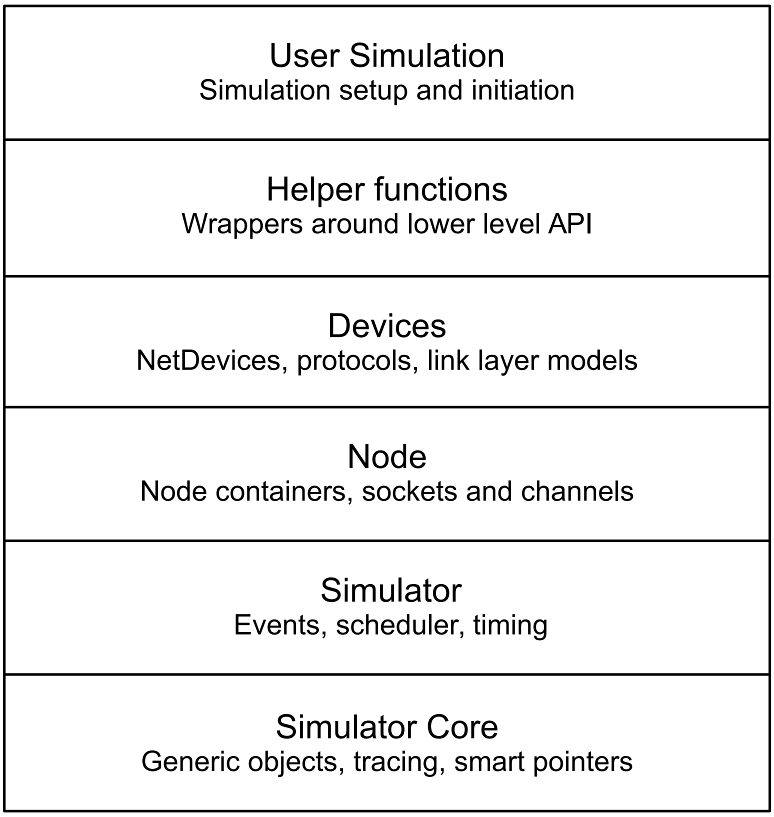

Figure 3.1: *ns-3* software structure overview

Application An application is an abstraction of a program that may be installed in a node to source or sink packet traffic. For example, an application may perform the regular transmission of UDP packets at prescribed time intervals.

NetDevice A NetDevice is an object that emulates a network interface card and when installed into a node supplies communication between the channel and node. As such the NetDevice will typically encapsulate MAC and PHY models. The NetDevice provides access to attributes concerning the link between the channel and node, such as the MTU size, event callbacks, and link statuses.

Channel A channel emulates the medium over which nodes in the simulated network topology communicate, such as a point-to-point Ethernet cable, or a channel in the 2.4 GHz ISM band. Channels have various attributes associated with them such as speed of the simulated transceivers, latency or free air propagation loss.

Figure 3.2 illustrates the composition of a simulation. There are multiple node objects connected to a shared channel for communication. A node consists of an application for traffic sourcing or sinking, a network that can perform addressing and routing, and a NetDevice for accessing the channel.

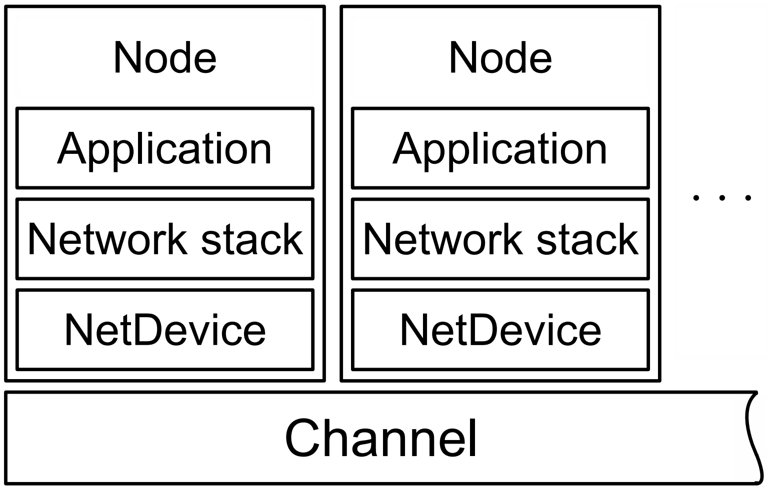

Figure 3.2: *ns-3* simulator object overview

3.2 Simulation Design

A simulation in *ns-3* is based upon the creation of nodes, and the installation of various communication channels, NetDevices, addressing, and applications. A typical simulation main program involves these stages:

- Nodes are created and grouped into containers.
- NetDevices are installed and nodes are added to the channel.
- The network stack is installed and nodes are assigned an address.
- Traffic generation and sinking applications are installed.
- Applications are scheduled to start and the simulation is scheduled to stop.

A simple UDP echo simulation scenario using an Ethernet interface works as follows. Nodes are created and a helper interface to a CSMA channel is used to add all nodes onto the simulated Ethernet network. The CSMA helper takes each node and adds a CSMA NetDevice, installs a unique MAC address, and attaches the node to a shared CSMA channel used by all when communicating. The Internet Protocol (IP) addressing, Transmission

Control Protocol (TCP) and User Datagram Protocol (UDP) layers are then installed by the internet stack helper and IP addresses are assigned by an address helper. Applications are installed on the UDP client and server to transmit UDP packets, and echo them back.

This simulation is then run for a certain period of time or until all the specified UDP packets have been sent by the originator. Various forms of output are available such as comprehensive logging via standard output, statistics computed by FlowMonitor [36] detailing each traffic flow, or trace files in PCAP format.

3.3 WLAN Simulation

ns-3 includes models for numerous MAC and PHY amendments of the IEEE 802.11 WLAN standard with varying degrees of completeness. It provides support for all the common 2.4 and 5 GHz PHY specifications, a single spatial stream and a single antenna. There is support for infrastructure, independent and mesh BSSs, all using standard DCF, EDCA, immediate and HT-immediate compressed block acknowledgements, aggregate MSDUs, and MPDUs. [1]

A simulation of an 802.11 network is created in the same manner as the Ethernet network detailed in Section 3.2. A lot of the upper software stack is agnostic to lower layers used to build and access the network. In the case of an 802.11 network only the objects below the Link Logical Control (LLC) layer are unique to this type of network. Specifically the NetDevice, channel access mechanisms and channel are unique.

The *ns-3* WLAN model is made up of various components specific to the model implementation as illustrated in Figure 3.3. Data for transmission is received by the WifiNetDevice from the network layer, which then prepends the LLC header to the packet and enqueues it with the upper MAC of the STA or AP. From here the packet is sent to DcaTxop if QoS is disabled, or to the EdcaTxop object that matches the traffic class for that particular packet. DcaTxop and EdcaTxop are similar objects that interact

[1] MPDU aggregation is currently a patch to the simulator [37] and as such may be subject to change before it is a part of the official release. The particular version used in later chapters is based on Patch Set 3 with further modification to fix numerous errors and support the presented validations.

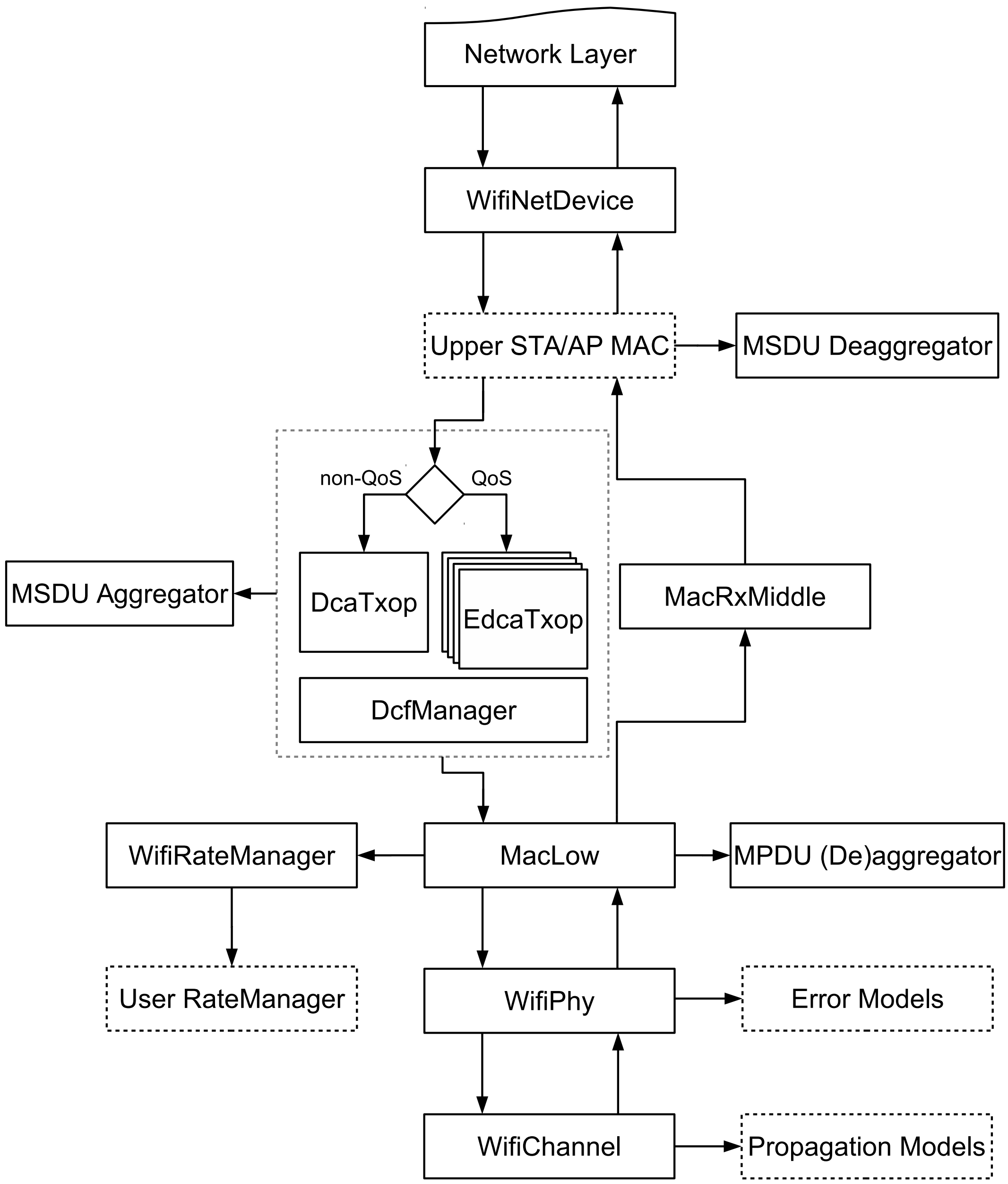

Figure 3.3: *ns-3* WLAN model structure overview

with the DCF and provide the bridge between the enqueueing of packets from the upper MAC and transmission of packets by MacLow. MacLow deals with sending data to and receiving data from the WifiPhy below, and handles transmission and reception of Control frames such as RTS/CTS sequences and acknowledgements. MacLow communicates with the WifiRateManager to establish the PHY rate to use in that packet. Each PHY rate is represented by a WifiMode object that holds the various parameters that characterise the rate. The WifiPhy passes the packet to the channel, which applies any propagation loss and delay calculations on it, and then forwards the packet to the WifiPhy of all other devices connected to the channel.

The receive path of a packet is notionally the reverse of the transmit. When a WifiPhy receives a packet from its connected WifiChannel it models receiver processes to establish whether this packet should be received correctly and passed up to MacLow. To do this it will take account of the received signal strength, interference due to other communicating nodes in the vicinity, current state, and bit and packet error rates.

Correctly received packets are forwarded up to MacLow which takes action based on the type of the packet. For frames requiring immediate acknowledgement MacLow will schedule transmission of the ACK frame. Data and management type frames have the 802.11 MAC header removed and are then forwarded up the stack through MacRxMiddle which handles duplicate detection and reassembling fragmented data frames.

The upper MAC can then forward data frames to the WifiNetDevice and onwards, or send the appropriate response to management frames. The upper STA MAC contains a state machine that deals with association and the necessary procedures surrounding that. The AP MAC handles Association Responses and Beacon scheduling.

ns-3 includes support for block acknowledgement and MSDU and MPDU aggregation. Only compressed HT-immediate block acknowledgements are fully implemented although there is partial support for HT-immediate as it is used for implicit block acknowledgements when responding to A-MPDUs. A block acknowledgement agreement is created through an ADDBA Request/Response sequence where various parameters pertaining to the agreement are exchanged and established. In *ns-3* this creates an OriginatorAgreement which keeps track of the current state. A BAR is made when the number of outstanding packets is equal to the window size or there are outstanding acknowledged packets but none pending transmission.

Both MSDU aggregation and MPDU aggregation are supported in *ns-3*. MSDU aggregation is performed by EdcaTxop when it is informed by the DcfManager that it has access to the medium. Packets of corresponding AC are built into an aggregate which is then forwarded down to MacLow for transmission. A-MSDU deaggregation takes place in the upper MAC before individual MSDUs are forward up to the WifiNetDevice.

MPDU aggregation occurs in MacLow when an initial packet is forwarded down. The A-MPDU is built from up to 63 additional MSDUs with the same traffic identifier and receiver MAC address, while ensuring the aggregate packet stays under the length limit. In order to model interference occurring during an A-MPDU transmission each MPDU is forwarded to WifiPhy as a separate transmission with all but the first lacking a preamble and PHY header. Deaggregation is simple for the receiving MacLow as it only needs to strip the subframe header and padding from the MPDU and forward the MPDU to the internal receive call as normal.

4

Simulation Model Design

As shown in Chapter 3 the *ns-3* WLAN model serves as an excellent base for the 802.11ad modifications as the core functionality already exists, simplifying the additions to only those introduced in the amendment. This chapter will present these changes in three distinct but related categories. Section 4.1 outlines how the antenna subsystem works and what provisions have been made to allow modelling various types. Section 4.2 explains the PHY modifications made to provide the raw data rates specified. The bulk of the work comes out in Section 4.3 where the beamforming systems and procedures are detailed.

Figure 4.1 gives a basic overview of the additions to the *ns-3* WLAN model. Minor objects and those not in the direct transmit and receive path been omitted from this figure. In order to support IEEE 802.11ad there have been several additions to the WLAN model:

- An abstract and a derived steerable antenna model has been implement for support of modelling directionality and directional gain.
- A Beacon transmitter object has been designed for sending multiple DMG Beacon frames.
- Modelling of the various beamforming procedures through a Beamforming Engine implementation.
- Creation of an extensible AWV Manager for evaluating beamforming algorithms.

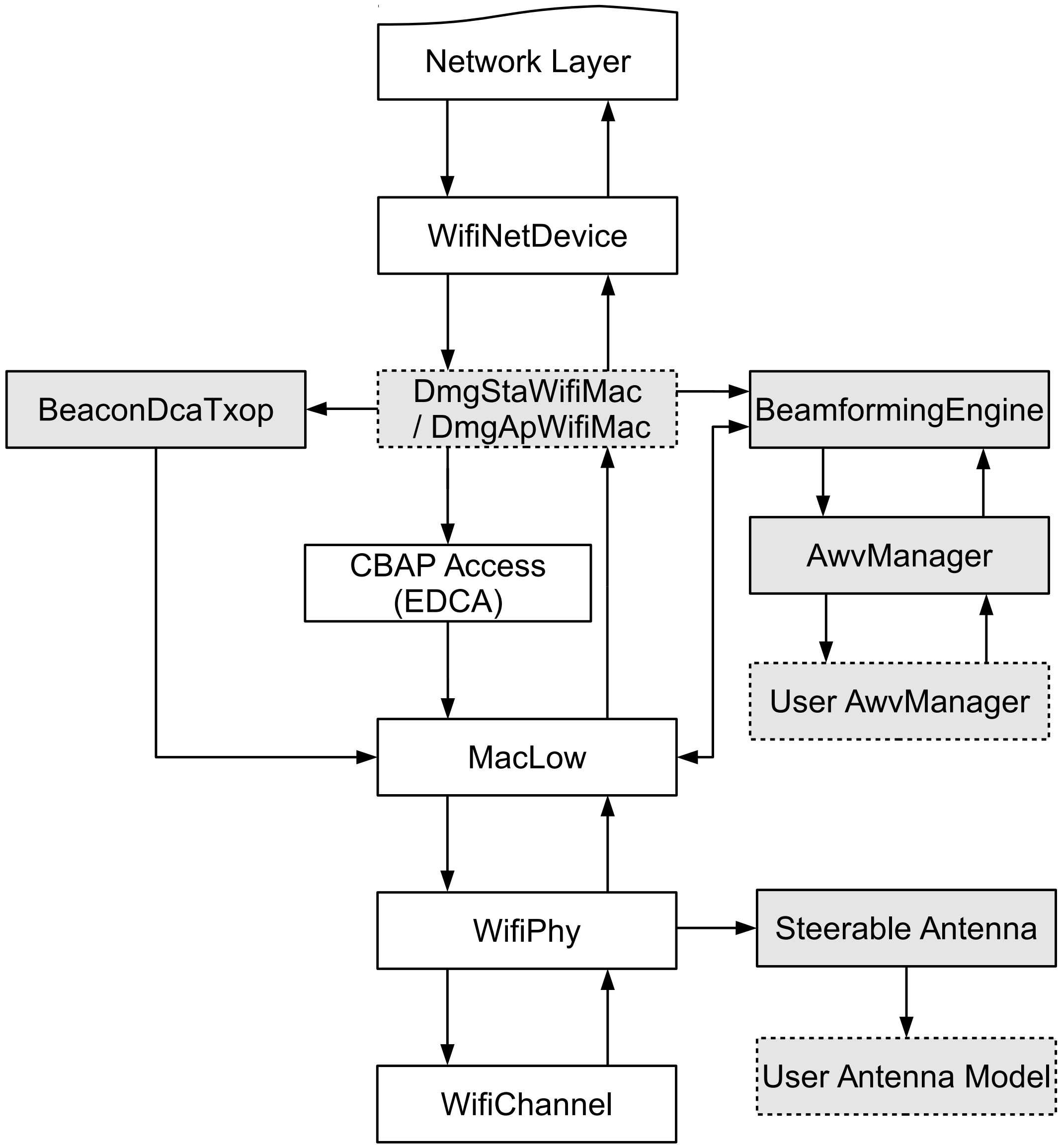

Figure 4.1: Overview of 802.11ad changes to the *ns-3* WLAN model

4.1 Antenna

ns-3 provides an extensible antenna module that allows the representation of various types of antenna. Antenna models inherit from a parent class in order for the user definable attribute system to work correctly. The only member function to be implemented needs to return the gain for that antenna. The existing antenna models were unsuitable for use in this work as they were not parameterised by direction, which is necessary for modelling and testing of the proposed MAC modifications.

To model directional parameterisation we designed an abstract steerable antenna model that allows a variable width radiation pattern to be moved within a two-dimensional space. This model has a definable number of sectors that the 360° coverage area is divided into, with the beamwidth of each sector covering a portion of the area. The PHY, and by extension the MAC, can pass in a value that sets the antenna boresight to a particular direction. The possible values range from 0.5 through to the number of sectors + 0.5. This direction abstraction allows simplification of the MAC design, as will be shown in Section 4.3.3.

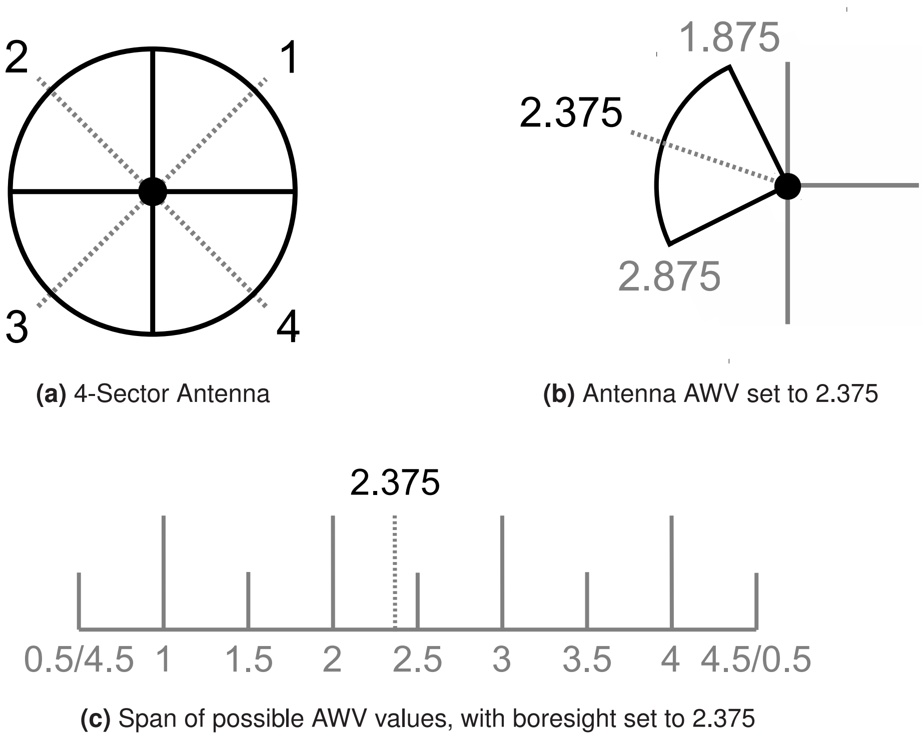

Figure 4.2: Steerable antenna representation

Figure 4.2a shows a 4-sector antenna model, where each sector has a 90° 3 decibel (dB) beamwidth. The AWV of the boresight for each sector corresponds to an integer value, while a non-integer is known as an AWV to the MAC. The possible positions range from 0° to 360°, or 0.5 to 4.5 in possible AWV values. The possible AWV space is shown in Figure 4.2c, while Figure 4.2b shows the AWV of the beam set at 2.375. When the PHY sets the antenna configuration using an AWV it is translated into a position in radians within the possible 360° space. An AWV of 0 is a special value which corresponds to an quasi-omni-directional configuration

A derived antenna model called Steerable60GhzModel was designed to model the gain based on the reference antenna model presented for IEEE 802.15.3c evaluation [38]. When the channel is given a packet to send to the other STAs in the simulation it computes the theoretical link margin using the transmitter's PHY power, the transmitter's antenna gain, and the path loss using a separate model. The packet is then sent to each receiver where their antenna and PHY gain are added on.

These antenna gains are based on the angle the nodes make to each other. The antenna model finds the gain to return through the angle the opposite node makes to the boresight of the beam set through the antenna configuration. If the angle made falls outside of the main radiation lobe then a separate side lobe calculation will take place. If the quasi-omni-directional configuration is applied then Steerable60GhzModel acts as a isotropic radiator with uniform gain in all directions.

4.2 PHY

Data transmissions in *ns-3* are represented as a receive delay which is a function of the number of bits in the transmission, type of PHY used, and various other transmission parameters. As part of the work described herein, all mandatory PHY rates have been implemented, and in addition the rates see the highest throughputs. These are the Control, SC, or OFDM PHY. The structure of a 802.11ad PPDU transmission is illustrated in Figure 4.3. To model this in *ns-3* TRN fields have been added for use in beam refinement and tracking. Every PPDU transmission consists of a preamble, PHY header, payload, and zero or more TRN and Automatic Gain Control (AGC) fields. The duration of the preamble and PHY header are constant between the 802.11ad bit rates but may vary between PHY types. The payload duration is a function of the payload size and the MCS used. The number of symbols or blocks is directly related to factors such as the code rate, constellation size and length. These in turn are used to calculate the duration of the payload based upon Control, SC, or OFDM PHY specific equations.

TRN and AGC fields are predefined Golay complementary sequences used for training. While the AGC fields are not used for Automatic Gain Control in the model, it is important to include these fields in order to ensure correct burst durations. In the model a both TRN and AGC occurs in a separate transmission to the PSDU and are modelled as an empty Packet object. Objects can identify the empty packet as a TRN field through accompanying metadata. A TRN field is a constant duration known by the WifiPhy as they have no length in bits.

In order to model the 802.11ad PHY rates, WifiMode objects for each MCS, and the transmit burst duration calculations defined in the amendment have been implemented.

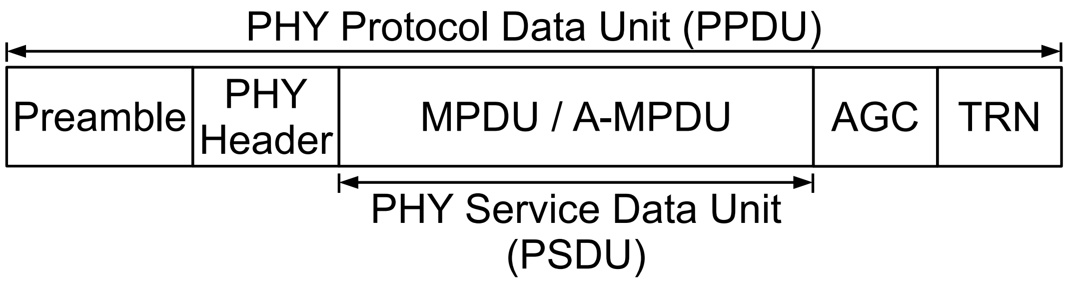

Figure 4.3: Logical structure of a transmission

4.3 MAC

In order to support the 802.11ad amendment, various parts of the MAC have been modified and extended. These modifications encompassed DMG Beacon transmissions, packet metadata, beamforming procedures, aggregation, and frame implementation. In particular:

- Implementing DMG frames and the necessary IEs.
- Updating rate selection to accept the DMG MCS structure.
- Design of a DMG Beacon transmitter object for sending multiple Beacon frames.
- Extending the packet metadata object to act like a PHY header.
- Modelling of the various beamforming procedures through the DmgBeamformingEngine implementation.
- Creation an extensible AWV Manager for evaluating beamforming algorithms.

MPDU aggregation has been added to the model using external draft modifications [37] (Patch Set 3). Although substantial bug fixing and validation of this code was required.

4.3.1 DMG Beacon Transmission

In a DMG AP multiple DMG Beacon frames are scheduled for transmission at each TBTT and they are transmitted across various sectors. DMG Beacon transmission acts as an initiator TxSS, and as such a mechanism that handles sending DMG Beacon frames has been designed. A BeaconDcaTxop object has been created which takes a template DMG Beacon frame from the upper AP MAC and transmits the requested number of DMG Beacon frames. For every transmission the sender gets a sector to use from the AWV Manager (see Section 4.3.4) and sets the applicable fields in the frame such as the SSW IE, which provides identification of the transmission antenna and sector.

Amongst all the information conveyed in the DMG Beacon frame, the structure of the beacon interval and A-BFT are the most important to any STA that is associated or wants to associate with the AP. The receiving STA needs to know at what point in the future the A-BFT and DTI begin, whether the A-BFT is a TxSS or RxSS, and the number of slots and the length of each slot.

If necessary, this information can then be passed to the Beamforming Engine (see Section 4.3.3) for when it begins contending in the A-BFT.

4.3.2 Packet Metadata

The IEEE 802.11 standard defines a MAC/PHY interface in the form of a RxVector and TxVector parameter set. The TxVector accompanies a MPDU sent to the PHY for transmission and supplies it with transmission parameters. Likewise, the RxVector accompanies an ingress PSDU and supplies the MAC with reception parameters.

An object of similar purpose current exists in *ns-3* for relaying transmission side parameters in relation to 802.11n. To support IEEE 802.11ad we have further extended this object as a replacement for the PHY header that is not present in the *ns-3* WLAN implementation. When a MPDU is transmitted the PHY performs various actions based on these fields, such as setting the antenna configuration and calculating the transmit duration. The channel takes the metadata alongside the packet and pushes both to all receiving STAs. The receiving STA builds a RxVector from the relevant parts of the TxVector and stores various parameters relating to how the packet was received, such as the receive antenna configuration and the calculated SNR. The MAC can use this information in decision making. For example, the Beamforming Engine can use the SNR to make decisions on AWV transmit and receive training.

4.3.3 Beamforming Engine

The Beamforming Engine performs the beamforming procedures and protocols with the guidance of the AWV Manager. The AWV Manager which handles most of the logic and control regarding the sectors and AWVs used in transmission and reception. Of the five types and stages of link training only the mandatory three have been implemented, the SLS, BRP phase and Beam Tracking.

The amendment does not define how a STA should determine what the best sector or AWV a frame was received on. In this work all decisions are based on the SNR as the means of comparing configurations as this is readily available and provides a good measure of signal performance. Although the framework designed does allow testing other approaches.

Sector-Level Sweep

The Sector-Level Sweep is the initial stage of beamforming training, with the goal of establishing communications are the CPHY rate. The SLS can occur in two places within the beacon interval: the A-BFT; and the DTI. While these both work to the same end, the execution is rather different and there are some limitations present in the A-BFT.

When the Beamforming Engine begins a SLS it fills a internal transmission queue with individual Sector Sweep frames, either up to the maximum number of frames allowed by an agreement made at the start of the SLS or until the AWV Manager informs it there are no more sectors to try in this session. The Beamforming Engine queries the AWV Manager to get the sector used in transmission. In a RxSS this sector is static for the duration of the sweep and is either an quasi-omni configuration or the best transmit sector if a TxSS has been performed in a previous SLS. In a TxSS, as the queue is being built, the Beamforming Engine queries the AWV Manager for the next sector to use.

Each SSW frame contains a countdown field which decrements with every transmission until the last frame is set to zero. As the receiver may not hear every frame thus it may not know that the last frame has been sent. To compensate for this on every successful reception a timer is updated to the expected end of the sweep and the start of the following sweep or feedback.

If a STA still has more sectors that have not been used in a SLS due to length restrictions then it can split that training across multiple sessions. The SNR-Report field in the SSW-FB IE is particularly important in a split TxSS as it allows the receiving STA to compare the feedback of sectors that were tested in different sweeps.

At the conclusion of the ISS or RSS the receiver transmits feedback to its peer that informs it of its best transmit configuration if a TxSS occurred, or just the SNR if a RxSS occurred. If an ISS concluded then feedback is a part of the SSW frames transmitted by the responder in the RSS. If the RSS has finished then there is a separate SSW-FB frame sent by the initiator. This whole sequence is shown in Figure 4.4.

A-BFT The A-BFT provides the initial link training and is primarily targeted at STAs that will try and associate to the AP. On reception of a DMG Beacon frame the STA informs the Beamforming Engine of the length of the A-BFT, number of frames allowed in a slot, and the type of the training to perform. Following the BTI where the AP transmits one or more DMG Beacon frames, in our implementation a STA wishing to associate will contend for a slot in each A-BFT until association is complete.

Once the SSW frame queue has been built the Beamforming Engine will contend for access by choosing a slot at random from 0 to the maximum slot number. The expected slot duration is calculated from the maximum frames per slot and transmission is then scheduled for the start of the slot. If a collision happens then the likely scenario is that the AP hears no discernible transmissions and no feedback will be sent to either STA, causing

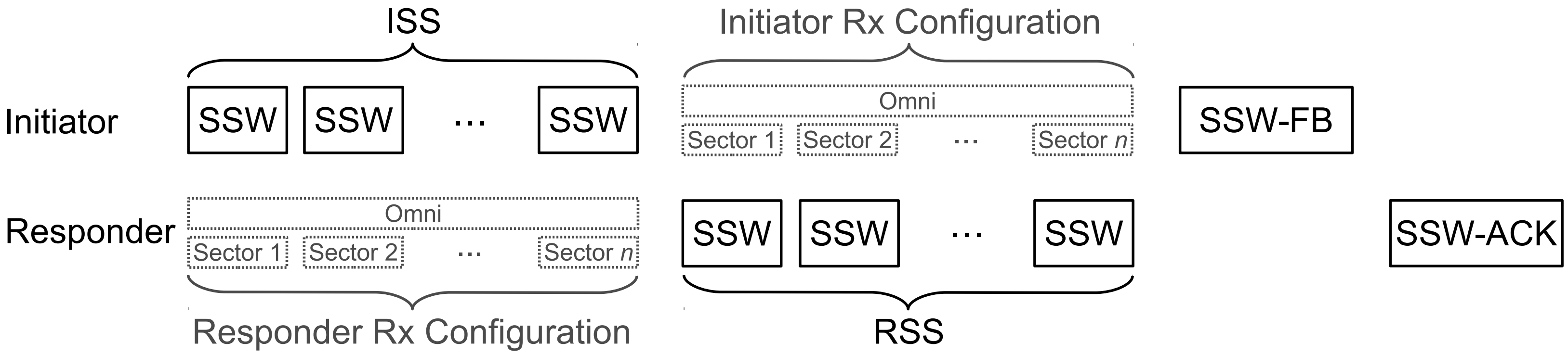

Figure 4.4: Structure of a Sector-Level Sweep

both to retry in a subsequent A-BFT session.

A slot has ended after an appropriate interframe spacing following the transmission of the SSW-FB frame. If there are more slots then the session will continue on, otherwise the DTI begins and the STAs are free to begin contention-based access.

The DmgStaWifiMac will not try to associate with an AP until both responder RxSS and TxSS have been completed in the A-BFT. This information is known to the AWV Manager as it keeps track of the sectors and sweeps performed.

DTI It is necessary for the SLS to be allowed in the DTI as the BTI/A-BFT training does nott allow for initiator receive training. A SLS in the DTI is requested by the upper STA MAC for a particular peer along with the type of ISS and RSS. The Beamforming Engine sends a Grant frame to the peer requesting training through the Beamforming Control IE. A Grant ACK response is sent by the responder which confirms and initiates the SLS. This sequence allows the responder to tell the initiator how many receive sectors it supports so that an initiator RxSS is possible.

Following this, the session operates as detailed above: an ISS and RSS occur and a SSW-FB frame is sent to the responder to provide feedback. After the feedback the responder transmits a SSW-ACK frame to the initiator which ends the SLS. These last two frames allow either party to request additional beamforming training through the BRP phase.

BRP

The purpose of the BRP phase is to further refine the beamformed link begun during the SLS. It is can consist of up to four parts that finish with a highly trained link. The four subphases are BRP setup, MID, BC, and BRP transactions. Only the mandatory two have been implemented; BRP setup and BRP transactions.

The BRP setup subphase allows the exchange of information relating to the following subphases. It may be omitted if the BRP transactions immediately follow a SLS. The Beamforming Engine currently does not support continuation of refinement following the SLS so the BRP phase always begins separately using the setup subphase.

There are three specialised methods within the Beamforming Engine that solely deal with the BRP phase: BeginBrp(); BrpManager(); and SendBrp(). The initiator's upper MAC asks the Beamforming Engine to begin BRP with a particular peer using BeginBrp(), which sets up various states and then calls the state machine manager - BrpManager() - to initiate the phase. BrpManager() pass a list of transmission parameters to SendBrp() which builds a BRP frame based on these parameters and after contention forwards the packet to MacLow for transmission. The responder's Beamforming Engine receives the packet and its BrpManager() responds accordingly.

Each BRP frame can be both a request and response. If the TxTrnReq field in the frame is set then transmit training is requested and the necessary number of TRN fields are appended to the end. A receive training request is made by setting the RxTrnL field to the number of TRN fields it would like to train on in a following BRP frame. A transmit response is made by including feedback to the peer which notifies it which TRN index was best, the originator then maps the index to privately held list of attempted AWVs. TRN fields are included in a receive training response and the receiving STA will test a receive AWV on TRN field.

In the simulation environment a TRN field is simply represented as an empty packet occupying a fixed quantity of time that is sent by MacLow. The presence and number of TRN fields is conveyed in the TxVector so the MAC and PHY are notified and can make allowances when calculating durations. Each TRN field is accompanied by a TxVector that allows identification.

The BRP setup subphase is a simple exchange where the initiator requests beam refinement with a responder and lets it know of its capabilities. The responder replies with its own capabilities, finishing the subphase. The next packet the initiator transmits will be a part of the BRP transactions and may have TRN fields appended if it is performing transmit training or the responder requested receive training during the setup. The BrpManager() uses the AWV Manager to get the transmitting and receiving AWVs to use, and also to store and utilise the feedback. The responder replies with its own BRP frame, performing receive or transmit training. Training continues in a back and forward manner until both STAs are no longer requesting anything. These frame exchanges are shown in Figure 4.5 where various training requests and responses are made by each STA.

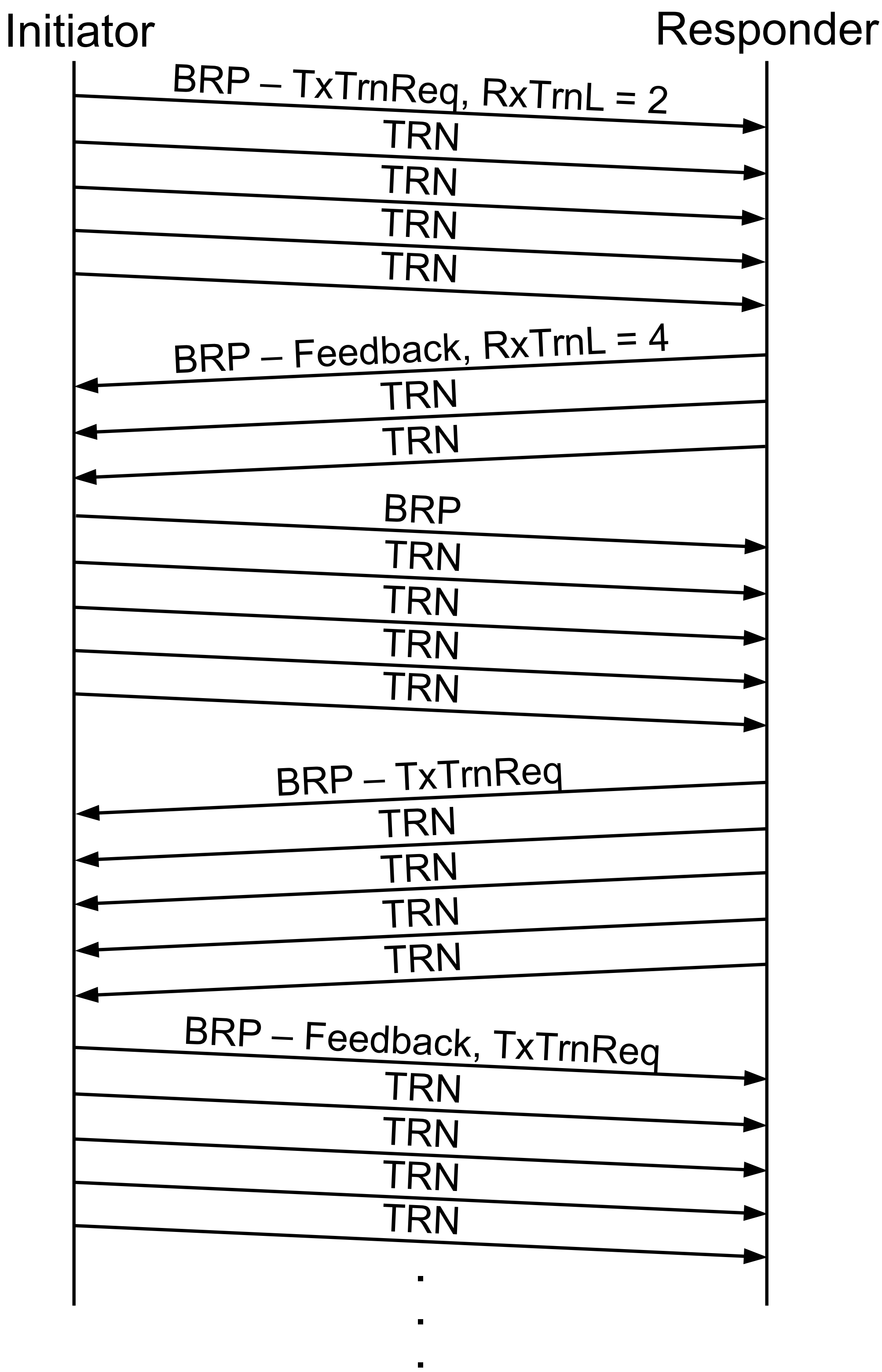

Figure 4.5: A sequence of BRP frame exchanges

Beam Tracking

Beam Tracking is used to continually adjust receive and transmit AWVs outside of a BRP session. During Beam Tracking, TRN fields are appended to regular packets and the appropriate fields in the TxVector are set. The upper MAC signals the Beamforming Engine to perform Beam Tracking to another STA in the next packet transmission. The Beamforming Engine queries the AWV Manager for a list of the receive or transmit AWVs to use depending on the type of training that the AWV Manager wants to perform. MacLow is then told that a request has been made so it can set the appropriate fields in the TxVector on the next packet for the target.

If transmit training is requested then the AWV list is passed to MacLow so it can transmit the TRN fields. The receiving STA is alerted to the presence of TRN fields by TxVector so it sets its receive antenna configuration to optimal if not done so already. At the conclusion of the PSDU transmission MacLow will iterate through the list of AWVs and transmit a TRN field for each one. The receiver's Beamforming Engine and AWV Manager pick the best transmit AWV that was received and build a BRP frame for relaying the feedback. This is given to MacLow so it can be appended to the end of the next frame back to the originator as a part of an A-MPDU.

Receive Beam Tracking works by asking the responder to append TRN fields to the next packet to the initiator by setting the appropriate fields and training length in the TxVector. The responder's MacLow matches egress packets to a receive training request and schedules TRN fields to be transmitted at the end of the PSDU. When the initiator receives a packet from the responder it is notified to the presence of TRN fields through the TxVector so it can begin testing receive AWV configurations at the conclusion of the PSDU.

4.3.4 AWV Manager

The purpose of the AWV Manager is to provide the Beamforming Engine with the values and decisions necessary for link training during the SLS, BRP and Beam Tracking. It acts as an interface for the retrieval of various learnt link parameters, the next parameters to test, and the next type of test to perform. The AWV Manager maintains a list of all the STAs that this STA has performed any beamforming procedure with. This information includes various link metrics and parameters, such as best receive and transmit AWVs for communicating with that STA, and the SNR for those AWVs.

The AWV Manager is made up of two parts that are designed to be extensible to allow modelling and testing of different types of AWV searching and feedback algorithms. DmgAwvManager is an abstract class provides an interface to the Beamforming Engine and contains various simple functions for the setting and retrieval of basic parameters. DmgBasicAwvManager is the currently designed derived class that calculates the information returned as part of the feedback processes, and informs the Beamforming Engine of the sectors and AWVs to use.

During a SLS the Beamforming Engine queries the AWV Manager repeatedly for the next transmit sector when it is building the SSW frame queue, or the next receive sector when it is switching receive configurations during training. DmgBasicAwvManager iterates through a list of sectors and returns the next appropriate receive or transmit sector.

Beam refinement involves more granularity than a sector approach and as such the AWV Manager operates differently. DmgBasicAwvManager takes an iterative refinement approach that progressively halves the search space until a configurable number of iterations have been reached. Beamwidths are not changed, only the boresight position. Using the best sector found as a centre, it starts with the maximum distance between sectors (1) and divides the span into a number of sections where the edges of each section correspond to an AWV to test. Feedback provides the next centre and search moves accordingly. The span is progressively halved each iteration until it is small enough that only minor movements and gains can be had. Beam Tracking uses a similar approach to beam refinement but keeps a fixed span size.

DmgBasicAwvManager bases feedback on the SNR of the received packet for all types of beamforming but in the future this could be extended to use the optional Channel Measurement IEs to provide more detailed information where BRP frames are concerned.

5

Simulation Model Validation

The 802.11ad model as described in Chapter 4 has support for very high throughputs in the 60 GHz band and antenna beamforming procedures used to increase link margin. These modifications need to be validated to ensure the network stack and the model are operating as intended.

Three simulation experiments were performed to prove the validity of the model in the three key modification areas: throughput; beamforming; and the antenna model. Throughput analyses were carried out, comparing theoretical performance to measured performance in UDP and TCP simulations involving a constant data stream between a pair of STAs. The validation of the beamforming procedures show that the designed algorithms can successfully find optimum antenna configurations to increase pairwise link margins using the designed antenna model.

The throughput statistics generated by the simulation model were provided using the FlowMonitor *ns-3* module [36]. This module tracks flows of traffic at the network layer and provides metrics such as bit rates and flow durations, and number of packets received and transmitted. Throughputs shown are an average taken over 20 simulation runs using the same transmission parameters outlined in the respective theoretical work. Only 20 simulation runs were used as the average variance between each run was less than 1% so little refinement would have been gained using any more. The conclusions we draw from these throughput analyses is that the 802.11ad model developed in this work is able to accurately and reliably simulate the 802.11ad amendment.

5.1 UDP Throughput Analysis

Simulation of UDP traffic is useful as it is easily modelled due to its basic design and the protocol does not rate limit itself as there is no flow control.. Throughput analysis of UDP traffic flowing between two nodes was carried out to prove the validity of the PHY implementation and the model's ability to handle large volumes of traffic. Calculations were completed to get an estimate throughput figure under well defined simulation conditions.

Figure 5.1: UDP simulation topology

As shown in Figure 5.1 the simulation scenario consists of an AP and a STA taking the roles of UDP traffic sender and receiver, respectively. Variable PHY rate was disabled in order to exclude variations in the rate selection algorithms and ensure consistent behaviour. All data transmissions were sent using the 6.756 Gbps OFDM PHY rate (MCS24) as it is the highest bit rate defined. Acknowledgement frames used the 1.155 Gbps SC PHY rate (MCS4) for standard-compliant operation. All UDP traffic used the Best Effort Access Category (AC_BE) with a AIFSN of 3 and with a minimum and maximum contention slot number of 0 and 15, respectively.

Each UDP packet is made up of 1472-octets of data payload, 8-octets of UDP header, 20-octets of IP header, making it 1500-octets long, with an additional 8-octets of LLC header prepended as the packet enters the 802.11 MAC. Both MSDU and MPDU aggregation was employed to increase MAC efficiency, up to the specified maximum size of 7935-octets and 262,143-octets, respectively, and HT-immediate block acknowledgements were implicitly enabled.

Assuming an always-full transmit queue, the maximum number of MSDUs capable of fitting in an A-MSDU is five. This gives 7622-octet A-MSDUs when the subframe headers and padding are taken into account. The

A-MPDU sent to the PHY contains a total of 34 A-MSDUs, each with a 26-octet MAC header, 4-octet Frame Check Sequence (FCS), and padding leading to a total size of 260,168-octets.

5.1.1 Calculation of Theoretical UDP Performance

A single UDP transmission sequence consists of several parts: a contention period; transmission of the UDP data; and acknowledgement. A STA begins access with a contention period in order to make sure that no other devices are using the medium and ensure it has fairly gained access. This is made of two pieces, the 18 μs AIFS, and the variable backoff period where the STA picks a slot from 0 to the upper window size and waits that many 5 μs slot periods. AC_BE has a minimum upper window boundary of 15 so the average slot number is assumed to be 7.5 with a normal distribution and the resulting average backoff time is 37.5 μs. In operation the upper boundary is subject to change due to internal and on-air collisions. These were not taken into account in the calculations as they are difficult to model and would be minimal as there are no competing BSSs.

The duration of this transmission using the OFDM PHY rate MCS24 is calculated using Equations 5.1, 5.2, 5.3, and 5.4. The transmission time is the sum of four separate durations: the Short Training Field (STF); the Channel Estimation (CE) field; the PHY header; and the duration of the data payload. The STF and CE durations are defined as 1.236 μs and 0.655 μs respectively, while the PHY header is 0.242 μs. The data duration is calculated from the length of the data field in bits and is dependent on the number of symbols used to transmit the codewords.

Using Equation 5.4 the number of codewords are calculated from the length of the PSDU in octets. R represents the code rate for the MCS in use, and L_{CW} represents the length of the each codeword, 672-bits.

The number of symbols is worked out in Equation 5.3 using the N_{CW} calculated in Equation 5.4 and N_{CBPS}, the number of coded bits per symbol. Both Equation 5.4 and Equation 5.3 output the ceiling integer value.

Finally, the number of symbols (N_{SYM}) multiplied by the symbol interval (T_{SYM}) of 0.242 μs, gives the data transmit time. For a transmission using MCS24 the code rate is 13/16 and N_{CBPS} = 2016. Therefore T_{Data} can be calculated for a 260,168-octet PSDU to be 307.582 μs, and the total transmit time of the PPDU is 309.715 μs.

$$TxTime = T_{STF} + T_{CE} + T_{Header} + T_{Data} \tag{5.1}$$

$$T_{Data} = N_{SYM} \times T_{SYM} \tag{5.2}$$

$$N_{SYM} = \left\lceil \frac{N_{CW} \times L_{CW}}{N_{CBPS}} \right\rceil \tag{5.3}$$

$$N_{CW} = \left\lceil \frac{Length \times 8}{L_{CW} \times R} \right\rceil \tag{5.4}$$

Following the data transmission is the 3 μs Short Interframe Space and the BA frame from the recipient. The BA is sent using the SC MCS4 PHY rate and uses different duration parameters and equations to the OFDM calculation. The STF and CE durations are the same but the PHY header has increased to 0.582 μs. The Equations presented in 5.5, 5.6, and 5.7 are used to calculate the PSDU duration given the MCS and length. The codeword length remains the same at 672-bits, ρ = 1, R is 3/4, and the number of coded bits per block (N_{CBPB}) = 448. The length of the BA is 32-octets which gives a total transmit duration of 3.09316 μs.

$$T_{Data} = (N_{BLKS} \times 512 + 64) \times 0.00057 \tag{5.5}$$

$$N_{BLKS} = \left\lceil \frac{N_{CW} \times L_{CW}}{N_{CBPB}} \right\rceil \tag{5.6}$$

$$N_{CW} = \left\lceil \frac{Length \times 8}{\frac{L_{CW}}{\rho} \times R} \right\rceil \tag{5.7}$$

Combining all of these factors the total time to transmit 260,168-octets is on average 371.30816 μs. FlowMonitor bases all measurements on the egress and ingress packet to and from the network layer. This means just the UDP payload, UDP header and IP header are tracked. As only 255,000-octets of the originators PPDU is counted as data payload the average theoretical throughput seen by the network layer is 5.494 Gbps, assuming ideal conditions. Figure 5.2 illustrates this sequence.

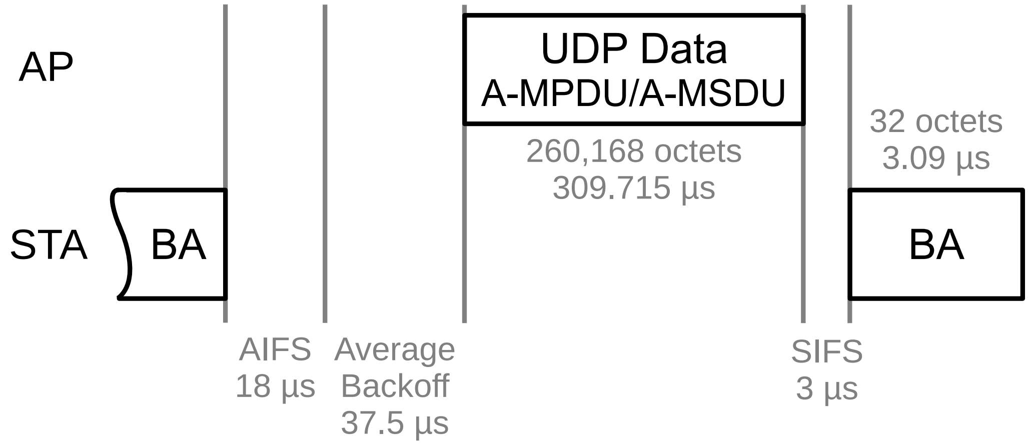

Figure 5.2: UDP data packet and acknowledgement transaction

5.1.2 Simulation of UDP Performance

As shown in Figure 5.3 an OnOffApplication transmitter application was installed on to the AP and a PacketSink application was installed in the STA. The packet generation frequency was such that the AP could always fill an A-MPDU with A-MSDUs up to their respective limits. Addressing was done with IPv4, the 802.11 Maximum Transmission Unit (MTU) was known to the host, and the Address Resolution Protocol (ARP) tables were pre-populated to reduce any potential overhead caused by address resolution.

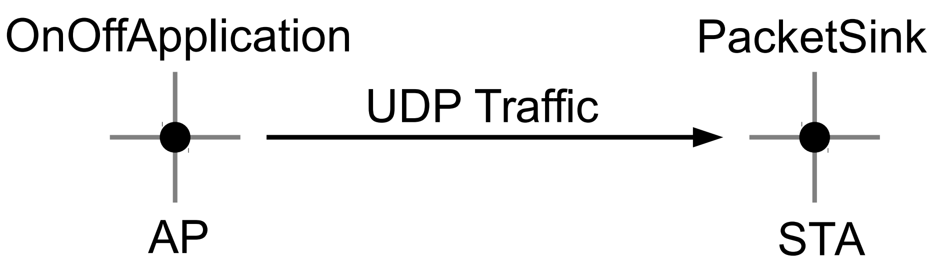

Figure 5.3: UDP simulation topology with installed applications

A total of 20 simulation runs were completed consecutively and an average was taken across the varying throughputs. Each simulation ran for a total of eleven seconds and the UDP sender began after one second. During the first second association and the beamforming procedures took place. The sender application was started after one second and finished when the simulation stopped on the eleventh second. The average throughput seen was 5.479 Gbps, 0.015 Gbps (-0.27%) less than the theoretical throughput.

Each beacon interval contains four DMG Beacon frame transmission and has an allocated time for the A-BFT. The total duration of these two periods is 412 μs. In 1 second there are generally 10 beacon intervals, so out of 1 second 4.12 milliseconds are spent not transmitting UDP data. Performing the theoretical analysis again but over 0.99588 second instead of 1 second gives a theoretical throughput of 5.471 Gbps. With Beacons and the A-BFT taken into account the simulated result is now better than the theoretical. This is most likely explain by the difference in average contention window between the theoretical model and simulated model. In the theoretical model it is assumed that the average contention window is exactly 7.5 slots as it is the mid-point between the upper lower bounds. This is most likely slightly lower than 7.5 in the simulation, resulting in quicker access and the increased throughput seen.

From the results seen here it can be concluded that UDP performance using the 802.11ad model designed is near enough to the expected results that it is valid. The cause of discrepancy between the expected and the actual result is most likely a slight difference in average contention windows.

5.2 TCP Throughput Analysis

TCP is a reliable, connection-oriented protocol that plays a significant role in the transport of a lot of today's network traffic. Analysis involving this protocol shows the expected throughputs of a 802.11ad network when used in bulk transfer scenarios such a file synchronisation and sharing, and other cases in need of a reliable, high-speed connection.

As shown in Figure 5.4 the simulation scenario consists of an AP and a STA taking the roles of TCP traffic sender and receiver, respectively. The TCP traffic sender generates and transmits TCP frames, and the receiver responds with TCP acknowledgements. All TCP frame transmissions use a constant MCS24 PHY rate, while 802.11 acknowledgements use MCS4.

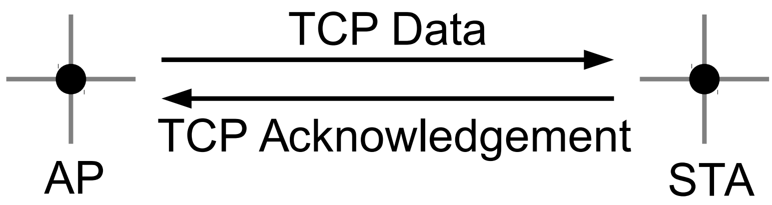

Figure 5.4: TCP simulation topology

AC_BE was used for all traffic and both types of MAC-level aggregation were employed. Only the TCP data sender had a BA agreement in place so all TCP ACK frames were acknowledged at the link layer using normal 802.11 ACK frames.

Keeping the same frame and aggregate sizes used in Section 5.1 the total TCP data PPDU consists of 34 MPDUs. Each MPDU contains 7622-octet A-MSDUs, making the total PSDU 260,168-octets long with headers and padding. Each A-MSDU carries 5 MSDUs and each MSDU is comprised of a 1460-octet data payload, 20-octet TCP header, 20-octet IP header, and a 8-octet LLC header. The only time a TCP option is used is during the connection setup for exchanging the window scaling parameters.

The BA frame is 32-octets and the normal ACK frame is 14-octets long. There are 170 TCP data frames carried in the originating A-MPDU so 85 TCP acknowledgement frames are sent back due to the delayed acknowledgement technique. Each TCP acknowledgement frame is 48-octets including IP and LLC headers, and when aggregated into an A-MSDU the total size is 5468-octets with delimiters and padding. It is more efficient for the TCP acknowledgement frame to be acknowledged at the link-layer with a normal ACK frame rather than a BA as this sequence is not long enough to compensate for the overhead introduced with a BAR/BA sequence [39].

Several constraints were put in place to make a comparison between theoretical and simulation throughputs valid.

- The theoretical calculations assume a steady TCP state where the congestion window has settled at its maximum value.
- The MAC-level transmit and reorder buffers at each end are assumed to be large enough to not overflow and drop packets.
- The TCP transmitter's contention window and receiver's receive win-

dow matches the maximum number of frames capable of fitting in a 262,143-octet A-MPDU given a 1500-octet data frame containing 1460-octets of TCP payload.

- The TCP delayed acknowledgement technique is active and acknowledging only every second TCP frame.
- Collisions were not taken into account as they should be almost non-existent in the steady state. This is because in a steady state there are precisely 170 MSDUs in the PSDU and that perfectly fills the TCP contention window. As the data transmitter cannot send any more frames until it receives a TCP acknowledgement the only chance of collision comes between the TCP acknowledgement and the first Beacon frame in the BTI.

5.2.1 Calculation of Theoretical TCP Performance

This theoretical analysis is based on the methods used in prior work to determine the performance enhancements of A-MPDUs and A-MSDUs [40].

The total time to transmit the assumed window size is given in Equation 5.8 and shown in Figure 5.5. The two components to this are the time required to transmit the TCP data frame (Equation 5.9) and the duration of the TCP acknowledgement frame (Equation 5.10). In both $T_{TotalTcpData}$ and $T_{TotalTcpAck}$ calculations AIFS is 18 μs, the backoff time (T_{bo}) is 37.5 μs, and SIFS is 3 μs.

Both TCP PPDUs in the exchange are transmitted at MCS24 while the BA and ACK frames are sent using MCS4. Using Equations 5.1, 5.2, 5.3, and 5.4 the transmit time for the 260,168-octet $T_{TcpData}$ PPDU is calculated as 309.715 μs, and the transmit time of the 5468-octet T_{TcpAck} PPDU is 8.667 μs. The MCS4 ACK frames are calculated with Equations 5.1, 5.5, 5.6, and 5.7 to be 3.09316 μs for both the BA and ACK frames as they take the same number of blocks (N_{BLKS}) to transmit.

$$TotalTxTime = T_{TotalTcpData} + T_{TotalTcpAck} \tag{5.8}$$

$$T_{TotalTcpData} = AIFS + T_{bo} + T_{TcpData} + SIFS + T_{BA} \tag{5.9}$$

$$T_{TotalTcpAck} = AIFS + T_{bo} + T_{TcpAck} + SIFS + T_{Ack} \quad (5.10)$$

The time to transmit 260,168-octets of TCP data is 436.562 μs including the spacings and various acknowledgement packets. The total number of octets that are seen by FlowMonitor across the entire TCP data PSDU is 255,000-octets. The total throughput is given by Equation 5.11 to be 4.672 Gbps.

$$Throughput_{Mbps} = \frac{255000 \times 8}{431.562} = 4,727 Mbps \quad (5.11)$$

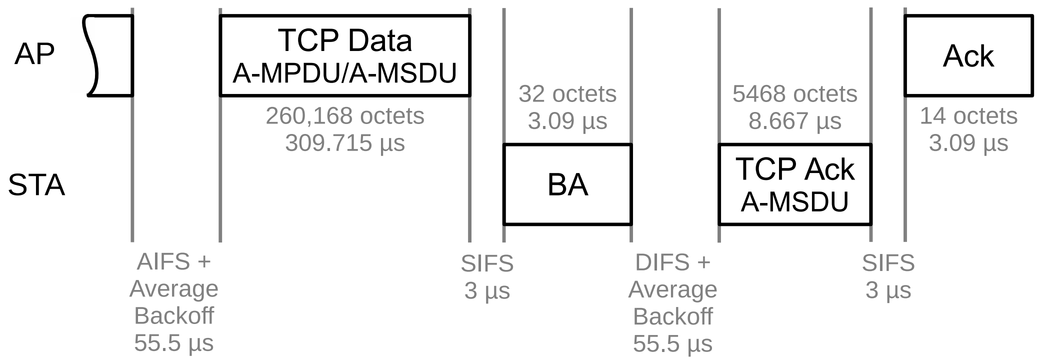

Figure 5.5: TCP data and ack transaction with link layer acknowledgements

5.2.2 Simulation of TCP Performance

As shown in Figure 5.6 an OnOffApplication transmitter application was installed on to the AP and a PacketSink application was installed in the STA. The packet generation frequency was handled through the built-in *ns-3* TCP implementation with regards to the transmitter congestion window and the receivers advertised receive window. The congestion-avoidance algorithm used was TCP New Reno [41]. The receive window was advertised using window scaling as 248,200-octets as this mapped to the maximum amount of TCP payload that could fit in a PSDU with aggregation[1]. Addressing was done with IPv4, the 802.11 MTU was known to the host, and the ARP tables were pre-populated to reduce any potential overhead caused by address resolution.

[1] Window scaling is currently a patch to the simulator [42] and as such is subject to change before it is a part of the official release. The particular version used is based on Patch Set 2.

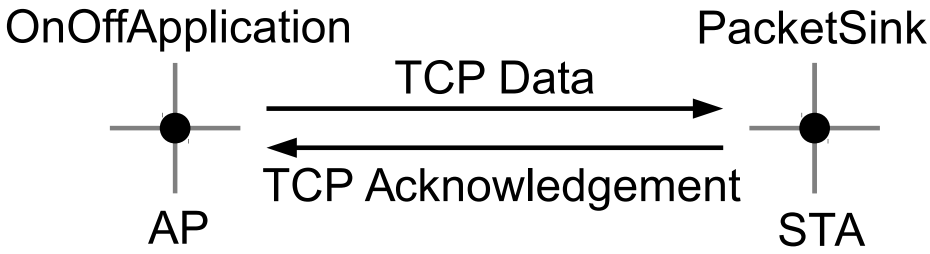

Figure 5.6: TCP simulation topology with installed applications

A total of 20 simulation runs were completed consecutively and an average was taken across the varying throughputs. Each simulation ran for a total of eleven seconds and the TCP sender began after one second. During the first second association and the beamforming procedures took place. The sender application was started after one second and finished when the simulation stopped on the eleventh second. The average throughput seen was 4.655 Gbps, 0.017 Gbps (-0.36%) less than the theoretical work.

Each beacon interval contains four DMG Beacon frame transmission and has an allocated time for the A-BFT. The total duration of these two periods is 412 μs. In 1 second there are generally 10 beacon intervals, so out of 1 second 4.12 milliseconds are spent not transmitting UDP data. Performing the theoretical analysis again but over 0.99588 second instead of 1 second gives a theoretical throughput of 4.653 Gbps. With Beacons and the A-BFT taken into account the simulated result is now better than the theoretical. This is most likely explain by the difference in average contention window between the theoretical model and simulated model. In the theoretical model it is assumed that the average contention window is exactly 7.5 slots. This is most likely slightly lower than 7.5 in the simulation, resulting in quicker access and the increased throughput seen.

From the results seen here it can be concluded that TCP performance using the 802.11ad model designed is near enough to the expected results that it is valid. The cause of discrepancy between the expected and the actual result is most likely a slight difference in average contention windows.

5.3 Beamforming Procedures

In this section the beamforming procedures and implementation used to find the optimal antenna configuration for receiving and transmitting packets are shown to be valid. In the simulation, two nodes are positioned at predetermined locations within a two-dimensional space. Calculations are performed to determine the angles to the positive x-axis in a Cartesian coordinate system that a line intersecting the nodes would make, allowing the optimal AWV values to be calculated for each node. These are compared and found to be in line with the values resulting from the beamforming procedures implementations in the model. It was assumed that multipath radiation would not affect results as no obstacles were present in the environment.

The two nodes are positioned as shown in Figure 5.7 where the Cartesian coordinates of the AP and STA are (-1, 1) and (2, -3) respectively, and a 90° beamwidth was used.

5.3.1 Calculation of Theoretical Antenna AWV

The horizontal distance between the nodes is 3 units, the vertical is 4, and the straight line distance based on Pythagoras' Theorem is 5. The angle the AP makes to the STA (θ_{AP}) using Equation 5.12 is calculated to be 306.87° relative to the x-axis in the first quadrant. Using Equation 5.13 the angle the STA makes to the AP (θ_{STA}) is shown to be 126.87°.

$$x = (\sin^{-1}(3/5)) + (3 \times 90^{\circ}) \tag{5.12}$$

$$y = (\sin^{-1}(3/5)) + (1 \times 90^{\circ}) \tag{5.13}$$

Based on the steerable antenna model created as part of this work the optimal AWV values can then be calculated using Equation 5.14. With a wide beamwidth of 90° the best AWVs for the AP and STA are $3.9096\overline{6}$ and $1.9096\overline{6}$ respectively. As there is no discrimination in the antenna model between receive and transmit configurations, these AWVs apply to both modes of operation.

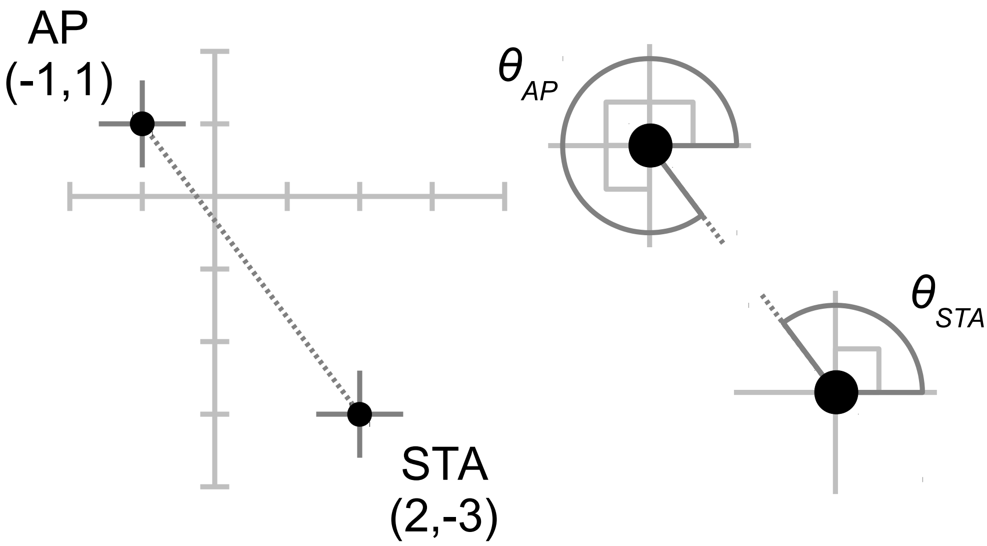

Figure 5.7: Layout of simulation nodes

$$config = \frac{angle - \frac{beamwidth}{2}}{beamwidth} + 1 \tag{5.14}$$

5.3.2 Simulation of Antenna Beamforming

Beamforming procedures occur over multiple iterations in various areas of the beacon interval, each working towards the end goal of a highly optimised antenna link margin. The first iteration is the BTI where the AP performs an initiator TxSS. Next, STAs wishing to associate with the AP use the A-BFT to perform a responder RxSS or TxSS as well as reporting back the best transmit sector used by the AP for communicating with them. Feedback is sent from the AP so the STA may know the best transmit sector to use. As the BTI is strictly an initiator TxSS the only place for the AP to perform receive training is during the SLS once the pair STA has associated. Following the SLS the BRP phase can occur, which results in a much more refined link.

Figure 5.8 and 5.9 show the stages the AP and STA go through while working towards this refined link, respectively. From feedback during a TxSS and its own internal measurements in a RxSS the AP initially decides that sector 4 is best when receiving and transmitting to the STA.

There are two configurable parameters that dictate how the BRP phase operates, the number of divisions to apply to the span length, and the number of iterations to perform before settling. For this work the number of divisions are four, resulting in five AWVs and TRN fields to test in each transaction, and the number of iterations was set to four. The centre of the initial search space is the best receive or transmit sector found during the SLS, 4, and the space is divided up and each boundary is tested. The best AWV of the first iteration is found to be 4 so the span is halved and the next five AWVs are tested. The phase continues in this pattern until all four iterations have been completed. The best resulting AWV in receive and transmit configurations is found to be 3.90625.

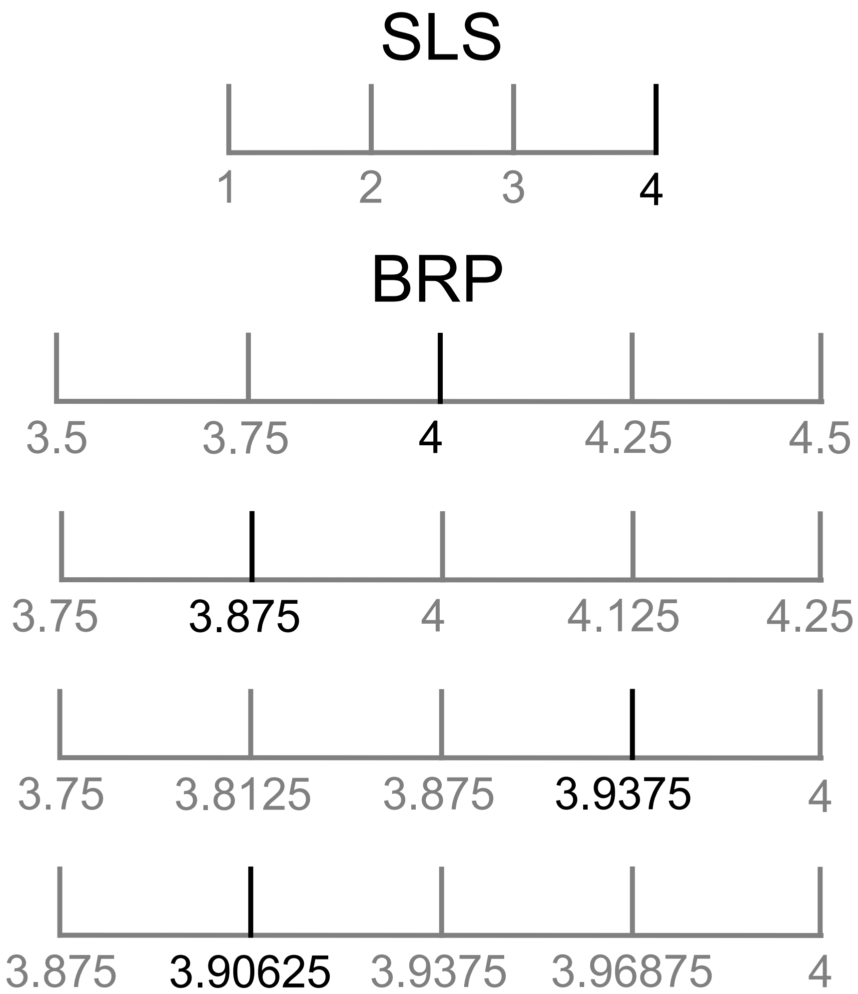

Figure 5.8: Sequence of AWVs tested against the STA during beamforming for the AP

Likewise, the various SLS phases result in sector 2 being picked as the best receive and transmit sector for the STA when communicating with the AP. The BRP phase is set up in the same way the AP is and the best receive and transmit AWVs converged on are 1.90625.

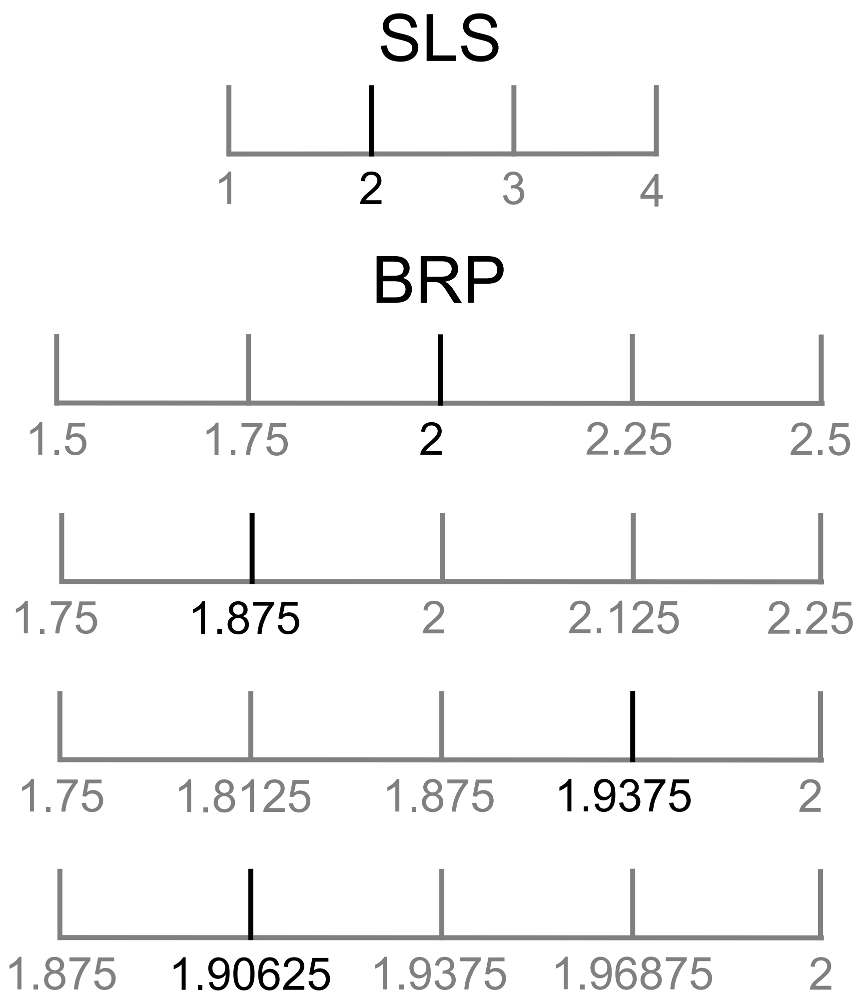

Figure 5.9: Sequence of AWVs tested against the AP during beamforming for the STA

Working from Equation 5.14 the angle of the AP to the STA with an AWV of 3.90625 is 306.5625° and the STA to AP angle is 126.5625°. The optimal angle of the AP to the STA found using the theoretical work is 306.87° and the STA to AP angle is 126.87°. For both nodes this is a difference of 0.3075° from the optimal angle to the simulation outcome.

Based on the antenna model presented for IEEE 802.15.3c simulation use [38], Equation 5.15 shows the maximum gain possible ($Gain_{max}$) with a beamwidth of 90° equals 7.18 dBi. The difference in power from the simulation to the theoretical AWV for both the AP and STA is 1.406×10^{-4} dBi based on Equation 5.16 and $\theta = 0.3075°$. These results show that the beamforming procedures implemented are capable of converging on a near perfect link margin when used to the fullest extent.

$$Gain_{max} = 20 \times \log_{10}\left(\frac{1.6162}{\sin\left(90^\circ/2\right)}\right) \tag{5.15}$$

$$Gain_{reduction} = 3.01 \times \left(\frac{2 \times \theta}{90^\circ}\right)^2 \tag{5.16}$$

$$Gain = Gain_{max} - Gain_{reduction} \tag{5.17}$$

6 Conclusion

While a lot of the physical aspects of the 60 GHz band are understood, [12] [14], there is a lack of application of the MAC processes and knowledge of how effectively an 802.11ad device operates. The goal of this thesis was to extend an existing network simulation environment to support the IEEE 802.11ad-2012 amendment in order to fully understand the impact that a directional, high-speed wireless link has on various parts of the networking stack and architectural design. *ns-3* was chosen as the network simulator to modify due to its open source nature, the quality of the existing WLAN model, and its performance when compared to other equivalent simulators.

In this thesis the *ns-3* WLAN simulation model is extended to support the 802.11ad MAC and PHY specifications. These changes are presented in Chapter 4 and include the:

- Design of an abstract and a derived steerable antenna model for support of modelling directionality and directional gain.
- Creation of a Beacon transmitter object for sending multiple DMG Beacon frames.
- Modelling of the various beamforming procedures through a Beamforming Engine implementation.
- Construction of an extensible AWV Manager for evaluating beamforming algorithms.
- Implementation of the Control, SC, and OFDM PHY rates and asso-

ciated transmit duration calculations.

In particular, a substantial amount of effort was spent on implementing the beamforming procedures. These procedures are necessary to deal with the highly attenuated and directional 60 GHz band. A steerable antenna model was designed and implemented to showcase and verify the beamforming procedures.

In Chapter 5 the presented WLAN model is validated through comparative analyses of the theoretical and simulated throughput measurements. UDP and TCP traffic flows were simulated and compared to theoretical throughputs, and the model was found to accurately represent achievable performance as shown in Section 5.1 and Section 5.2. The various beamforming procedures were found to converge on a near ideal orientation against an opposing device when compared to those found during a calculation of optimal positions in Section 5.3.

As shown above, this work has achieved its objective of accurately modelling the IEEE 802.11ad amendment through MAC and PHY extensions to the *ns-3* WLAN simulation model. These modification were validated and found to correctly match the expected results.

6.1 Future Work

There is a small amount of work left before these changes are ready to be folded into the main *ns-3* repository. Some of this work includes code tidying and consolidation, finish code commenting, doxygen writing, and writing regression tests.

There are numerous opportunities for further extensions to the designed model to enhance its operation and capabilities, some of these are listed here. Beamforming with non-static devices presents some challenges as to the logic and feedback mechanisms necessary for deciding when to re-initiate beamforming procedures and what type of procedures are required. Advanced channel feedback during BRP subphases would allow for greater decision making capabilities and move the onus of transmit AWV selection from the receiver to the transmitter. Moving to a 3-dimensional antenna model allows for more realistic and advanced simulations.

References

[1] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 2012.

[2] IEEE 802.11n: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, 2009.

[3] IEEE 802.11ac: Enhancements for Very High Throughput for Operation in Bands below 6 GHz, 2013.

[4] ITU-R P.676-10: Attenuation by atmospheric gases, 2013.

[5] L. S. Brakmo and L. L. Peterson. Experiences with Network Simulation. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '96, pages 80–90, New York, USA, 1996. ACM.

[6] A. R. Khan, S. M. Bilal, and M. Othman. A Performance Comparison of Network Simulators for Wireless Networks. Technical report, Computing Resource Repository, 2013.

[7] IEEE 802.11ad: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Very High Throughput in the 60 GHz band, 2012.

[8] Network simulator ns-3. http://www.nsnam.org/, 2009. Online; accessed March 2013.

[9] Network simulator ns-2. http://www.isi.edu/nsnam/ns/, 1997. Online;

accessed March 2013.

[10] T. R. Henderson, M. Lacage, and G. F. Riley. Network Simulations with the ns-3 Simulator. In *ACM SIGCOMM'08*, Washington, USA, August 2008.

[11] M. Lacage and T. R. Henderson. Yet Another Network Simulator. In *Proceeding from the 2006 Workshop on Ns-2: The IP Network Simulator*, WNS2 '06, New York, USA, 2006. ACM.

[12] IEEE P802.11 Working Group for Wireless Local Area Networks (WLANs). IEEE 802.11-09/0334r8 - Channel Models for 60 GHz WLAN Systems. Submitted to standards group, May 2010.

[13] Z. Genc, B. L. Dang, J. Wang, and I. Niemegeers. Home Networking at 60 GHz: Challenges and Research Issues. *Annals of Telecommunications*, 63(9-10):501–509, 2008.

[14] X. Tie, K. Ramachandran, and R. Mahindra. On 60 GHz Wireless Link Performance in Indoor Environments. In *Proceedings of the 13th International Conference on Passive and Active Measurement*, PAM'12, pages 147–157, Berlin, Germany, 2012. Springer-Verlag.

[15] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall. Augmenting Data Center Networks with Multi-gigabit Wireless Links. *SIGCOMM Comput. Commun. Rev.*, 41(4):38–49, 2011.

[16] IEEE 802.11a: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer in the 5GHz Band, 1999.

[17] IEEE 802.11b: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension In the 2.4GHz Band, 1999.

[18] IEEE 802.11g: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension In the 2.4GHz Band, 2003.

[19] IEEE 802.11e: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Quality of Service Enhancements, 2005.

[20] S. Ray, D. Starobinski, and J. B. Carruthers. Performance of Wireless Networks with Hidden Nodes: A Queuing-theoretic Analysis. *Computer Communications*, 28(10):1179–1192, June 2005.

[21] M. Gast. *802.11 Wireless Networks The Definitive Guide*. O'Reilly Media Inc., California, USA, 2nd edition, 2005. ISBN 0596100523.

[22] IEEE 802.1X, IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control, 2010.

[23] IEEE 802.15.3c: Millimeter-wave-based Alternative Physical Layer Extension, 2009.

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

[25] Federal Communications Commission. Millimeter Wave Propagation: Spectrum Management Implications. Technical report, Office of Engineering and Technology, Washington, USA, July 1997.

[26] D. S. Makarov, M. Y. Tretyakov, and P. W. Rosenkranz. 60-GHz oxygen band: Precise experimental profiles and extended absorption modeling in a wide temperature range. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 112(9):1420–1428, June 2011.

[27] C. R. Anderson, K. Bae, A. Verstak, N. Ramakrishnan, W. H. Tranter, C. A. Shaffer, and L. T. Watson. In-building wideband multipath characteristics at 2.5 & 60 GHz. In *Proc. IEEE 56th Vehicular Technlology Conference*, pages 97–101, September 2002.

[28] P. Smulders and L. Correia. Characterisation of propagation in 60 GHz radio channels. *Electronics & Communication Engineering Journal*, 9(2):73–80, April 1997.

[29] C. H. Doan, S. Emami, D. A. Sobel, A. M. Niknejad, and R. W. Brodersen. Design Considerations for 60 GHz CMOS Radios. *IEEE Communications Magazine*, 42(12):132–140, December 2004.

[30] D. M. Pozar. *Microwave Engineering*. John Wiley & Sons, Inc., USA, 4th edition, 2011. ISBN 9780470631553.

[31] D. Kam, D. Liu, A. Natarajan, S. Reynolds, H. Chen, and B. Floyd. Ltcc packages with embedded phased-array antennas for 60-ghz communications. *IEEE Microwave and Wireless Components Letters*, 21(3):142–144,

March 2011.

[32] A. Natarajan, S. Reynolds, M. Tsai, S. Nicolson, J. Zhan, D. Kam, D. Liu, O. Huang, A. Valdes-Garcia, and B. Floyd. A Fully-Integrated 16-Element Phased-Array Receiver in SiGe BiCMOS for 60-GHz Communications. *IEEE Journal on Solid-State Circuits*, 46(5):1059–1075, May 2011.

[33] C. A. Balanis. *Antenna Theory: Analysis and Design*. Wiley-Interscience, New Jersey, USA, 3rd edition, 2005. ISBN 9780471667827.

[34] S. Jansen. *Network Simulation Cradle*. PhD thesis, University of Waikato, 2008.

[35] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous. Direct Code Execution: Revisiting Library OS Architecture for Reproducible Network Experiments. In *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, CoNEXT '13, pages 217–228, New York, USA, 2013. ACM.

[36] G. Carneiro and M. Fortuna, P.and Ricardo. FlowMonitor: A Network Monitoring Framework for the Network Simulator 3 (NS-3). In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, pages 1–10, Brussels, Belgium, 2009. ICST.

[37] G. Badawy. Bug Issue 14549044: Adding MPDU aggregation. https://codereview.appspot.com/14549044/, 2013. Online; accessed December 2013.

[38] IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs). IEEE 802.15-06/474r0 - Reference antenna model with side lobe for TG3c evaluation. Submitted to standards group, November 2006.

[39] E. Perahia and R. Stacey. *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge University Press, Cambridge, UK, 1st edition, 2008. ISBN 9780521885843.

[40] B. Ginzburg and A Kesselman. Performance analysis of A-MPDU and A-MSDU aggregation in IEEE 802.11n. In *IEEE Sarnoff symposium*, pages 1–5, New Jersey, USA, 2007.

[41] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 2582 (Experimental), 1999. Obsoleted by RFC 3782.

[42] T. A. N. Nguyen. Bug Issue 24900043: TCP Window Scale and Timestamps Implementation. https://codereview.appspot.com/24900043/, 2013. Online; accessed December 2013.