Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science

Master Thesis

Combined Link Adaptation and Admission Control for More Reliable Groupcast according to 802.11aa

submitted by
Bruno Ranieri

submitted
June 10, 2010

Supervisor
Prof. Dr.-Ing. Thorsten Herfet
Advisor
Dipl.-Ing. Jochen Miroll

Reviewers
Prof. Dr.-Ing. Thorsten Herfet
Prof. Dr. Philipp Slusallek
Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, June 10, 2010

Unterschrift/Signature
Abstract

In this work a mechanism for physical link adaptation in wireless IEEE 802.11 networks for multicast transmissions is presented. The link adaption is based on the More Reliable Groupcast (MRG) Block Acknowledgment proposed in IEEE 802.11aa (draft) and on the signal-to-noise ration measured on the wireless channel. An implementation of the approach using the ns-3 network simulator is described. In experiments the achieved enhancement is demonstrated. In addition to this different admission control approaches for wireless networks are presented.

In wireless IEEE 802.11 networks multicast transmissions like audio/visual applications are transmitted without any feedback. To ensure correct transmission, the data is sent in the slowest but most robust encoding. This is a waste of time in most of the cases because the clients would also receive with a faster but not so robust encoding. The link adaptation controls the used encoding by the feedback provided from the receiving station. To overcome the lack of a feedback channel the IEEE proposes the MRG Block Ack in 802.11aa (draft).

Audio/visual applications are becoming more and more important in Internet today. For streaming media transmission like IPTV the requirements differ in the basic principle. While classic web applications do not have timing requirements but require complete and correct transmission, the opposite is true for audio/visual applications. They require strict timing conditions but tolerate some loss of data. In addition most audio/video applications are intended to be sent as multicast transmission, from one transmitter to several clients. To ensure that the audio/video applications requirements are met admission control, schemes are used which limit the allowed traffic on the network.
Title: Combined Link Adaptation and Admission Control for More Reliable Wireless Groupcast according to IEEE 802.11aa

For audio-visual multicast provided via a Quality of Service (QoS) access point, 802.11aa defines More Reliable Groupcast (MRG) with Block Acknowledgements.

A new client may attempt to join such a multicast group at any time, whereas the link quality of this new client is unknown at that point. If admitted without control, transmission quality could be degraded beyond the point where it becomes unreasonable to service the transmission at all. The adaptation of the 802.11a/g transmission rate to the combined link quality of the group is thus also a problem of client admission control. On the other hand, the problem of stream admission control is whether to admit a new stream to the network or not.

This thesis is aimed at elaborating and simulating a multicast client admission control scheme that shall consider the link quality as well as the mean bit rate and standard deviation of the respective A/V. The channel holding time when adapting the transmission rate to the link quality of a subset of the clients present in a multicast group shall be estimated in advance.

In this thesis the following tasks are to be solved:

- Description of block acknowledgement, admission control and multicast group management in 802.11 networks as well as post-AC schemes as developed at the lab.
- Brief description of 802.11a/g modulation and coding as well as 802.11e enhanced contention based channel access and derivation of channel holding times for frame transmission.
- Summary of rate adaptation mechanisms as found in the literature.
- Implementation of Multicast Block ACK according to MRG in ns-3 and performance evaluation in fading channels.
- Implementation of an SNR-based, feedback driven MRG rate adaptation mechanism with admission control using the weakest receiver for adaptation.
- Demonstration of cases were admission of some weak receiver(s) may reduce the quality in the group and the performance of the network.

Betreuer: 
Dipl.-Ing. Jochen Miroll

Gutachter: 
Prof. Dr.-Ing. Thorsten Herfet
# Contents

1 Introduction  13

1.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
  1.1.1 Audio/Visual Applications in today’s Internet . . . . . . . . 13
  1.1.2 Requirements for Audio/Visual Applications . . . . . . . . . 14

1.2 Problems . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

1.3 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2 Background  17

2.1 Network design . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
  2.1.1 The OSI Reference Model . . . . . . . . . . . . . . . . . . . . 17
  2.1.2 The TCP/IP Reference Model . . . . . . . . . . . . . . . . . . 18

2.2 IEEE 802 Protocol Architecture . . . . . . . . . . . . . . . . . . . . 19
  2.2.1 Architecture and Service . . . . . . . . . . . . . . . . . . . . . 19
  2.2.2 Other IEEE 802 Standards . . . . . . . . . . . . . . . . . . . . 20

2.3 IEEE 802.11 Wireless Network . . . . . . . . . . . . . . . . . . . . . 21
  2.3.1 Architecture . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
  2.3.2 Frame Types . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
  2.3.3 Medium Access Control . . . . . . . . . . . . . . . . . . . . . . 22
  2.3.4 Acknowledgment and Block Ack Mechanism . . . . . . . . . . 28
  2.3.5 Radio Propagation and Rayleigh fading Channel . . . . . . . . 31
  2.3.6 Modulation and Coding in IEEE 802.11a . . . . . . . . . . . . 33
8.1.1 Network Topology .............................................. 103
8.1.2 Wireless Channel ............................................. 104
8.1.3 Traffic Setup .................................................. 105
8.2 Tests Using MRG Block Ack .................................. 107
  8.2.1 Achieved Improvement ........................................ 107
  8.2.2 Treatment Of Weak Receives ................................. 109

9 Conclusion .......................................................... 117
  9.1 Achieved Enhancement .......................................... 117
  9.2 Simulation Accuracy ............................................ 118
  9.3 Further development ............................................ 119
Chapter 1

Introduction

1.1 Motivation

The Internet is replacing more and more the classic communication networks like the telephone network. It develops into a bidirectional network not only for text-based messages but also for phone calls and video conferences. An application which is becoming important nowadays, is the transmission of digital television over the Internet. The simplicity and reduced costs are the main reason for this, if one only has to run one cable to a customer instead of three for telephony, cable television and Internet.

1.1.1 Audio/Visual Applications in today’s Internet

Telephony over the Internet, referred to a Voice over IP (VoIP), today is a mass market product. Internet Service Provider (ISP), usually offer access to the Internet bundled with an VoIP account. A small device at the customers place manages the connection and offers a connection to plug in the present analog telephone.

Today streaming Media over the Internet has become more common today. Digital television is not only broadcast via satellite, cable or terrestrial radio broadcast but also via the Internet. In addition, user generated content plays an enormous role: In 2007 Youtube comprised 10% of all Internet traffic [1].

Another important market are online games, which have strict requirements with respect to timing in order to provide interactive game-play with other players over the Internet.
1.1.2 Requirements for Audio/Visual Applications

One important challenge in digital streaming media is the distribution. While the classic protocols like the Transmission Control Protocol (TCP) have nearly no requirements with respect to timing, its purpose is to ensure that all data is correctly transmitted. For audio/visual transmission the opposite is true. These transmissions can tolerate some loss of data during the transmission but have strict requirements with respect to timing.

The applications can be grouped into three major categories. Unidirectional, receive only: Applications like Internet Protocol Television (IPTV) or Internet-Radio. Unidirectional, with trick modes: Video on Demand (VOD) belongs to this category. The client can request a specific content to be sent. Bidirectional, interactive: VoIP, video conferences or online games.

Another difference compared to the classic transmissions is the approach used to reduce the amount of transmitted data if more than one client requests the same data.

ISPs provide so called proxy-servers in order to reduce the amount of web-traffic. The client can send requests not only directly to the server, but also to the proxy-server. If some user has requested the document before, the proxy-server has stored a copy of it and sends this to the new client. Otherwise the proxy-server requests the document from the server, stores a copy and sends it to the client. The ISP saves significant network traffic and money using this technique.

For audio/visual transmissions it is normal that more than one client wants to receive the transmission at the same time. The solution to send the data to every client independently would take an enormous amount of network traffic and corresponding costs. The solution is to emulate the classic radio broadcast transmission on the Internet with one sender and many receivers.

The two concepts in this context are broadcast- and multicast-transmission. Broadcast means that a packet is sent to all computers in a network. Multicast means that a packet is sent to a defined group of computers. The computers in this group do not need to be all in the same network but of course the networks must be connected with each other. The idea is to send each packet only once and duplicate it on its way through the network if the receivers are in different networks.

1.2 Problems

In the context of audio/visual application the timing of the transmission over the network is an important requirement. Thus the application defines an upper limit how long a station can wait until it must send the next data to fulfill the requirements.
Shannon’s noisy channel coding theorem describes the maximum capacity of a channel and the maximum possible efficiency of error-correction against different levels of interferences like noise and corruption.

In the IEEE 802.11 wireless network standard multiple data rates in which the data is encoded and sent onto the wireless channel, are defined. The data rates trade off robustness against corruption during the transmission against the time it takes to transmit the data over the wireless channel. The more robust a code is, the longer it takes to transmit the same amount of data.

The wireless channel is a shared medium. In an IEEE 802.11 wireless network only one wireless station can send data at a time. If one station is sending all other stations must wait or they will disturb the correct transmission of the sending station.

Even in the case that the link in the IEEE 802.11 is chosen in an optimal way, the available bandwidth of the wireless channel is limited. This sets an upper bound to the number of different audio/visual streams that can be transmitted at the same time and the problem is how to deal with this fact.

1.3 Outline

This section provides an outline on the organization of the thesis. The chapter Background gives an overview on the general design of local area networks according to the IEEE 802 protocol architecture. The focus is on the IEEE 802.11 wireless network. A brief description of the modulation and coding according to IEEE 802.11a/g and the EDCA medium access defined in IEEE 802.11e is given. Block acknowledgment for unicast transmission as defined in IEEE 802.11 is described to provide the basis for the MRG Block Ack described in the next chapter The IEEE 802.11a Amendment. A brief description of the extensions to the Internet Protocol for support of multicast traffic and the interrelationship between multicast and group addressed traffic is provided.

In the chapter The IEEE 802.11a Amendment the new block ack variant for group addressed traffic is presented. The so-called more reliable groupcast MRG Block Ack provides the feedback that is used by our implementation of a SNR-based link adaption for group addressed traffic.

The chapter Link Adaptation presents a summary of link adaption mechanism as found in the literature. In the chapter Admission Control the traffic stream mechanism that is defined in the IEEE 802.11 standard is presented. In addition to this a description of admission control schemes developed at the telecommunications lab is given.

The implementation of our SNR-based link adaptation is based on the ns-3 network simulator. A brief introduction to the simulator is given in the chapter on the ns-3.
network simulator. In the first part of the chapter Implementation a description of the changes needed in ns-3 is given. The second part gives the description of the implementation of our SNR-based link adaptation and the MRG Block Ack mechanism.

In the chapter Experiments the achieved results using our implementation are presented, with a focus on the achieved improvement compared to the legacy situation and the treatment of weak receivers that may reduce the performance in the network.

In the last chapter a summary and a prospect on future research questions that arouse from this work and upcoming technology is given.
Chapter 2

Background

In this chapter an overview of computer network design is provided. With a focus on the technical design and the parts needed to be known for an understanding of the IEEE 802.11aa amendment.

2.1 Network design

To reduce complexity computer networks are organized as a stack of layers. Each layer provides a certain service to the adjacent layers and works independently from the other layers. With this encapsulation the implementation of each layer is independent from the others and a layer can be implemented adequate for different kinds of computer networks.

2.1.1 The OSI Reference Model

The ISO OSI (Open System Interconnection) model [24] is shown in Figure 2.1. We call it the OSI model for short. It is grouped in seven layers, which can be briefly described as follows.

The upper four layers are implemented exclusively at the computers at the edge of the network. They provide user interface and control the end-to-end transportation between the computers.

**Application Layer** is the network interface for the applications that the user interacts with.

**Presentation Layer** translates the local data encoding to standard network interchange format.

**Session Layer** manages the connections between the computers.
CHAPTER 2. BACKGROUND

Figure 2.1: The OSI Reference model

**Transport Layer** provides the transport of data for the connections at the session layer.

The lower three layers are implemented at all nodes in the network. They manage the transportation through the network.

**Network Layer** manages the route the data takes through the different intermediate subnetworks.

**Data Link Layer** defines how a subnetwork is managed.

**Physical Layer** defines how the data is encoded to and from the physical medium.

2.1.2 **The TCP/IP Reference Model**

The predecessor of the Internet was the ARPANET. This network was based on a slightly different reference model, the TCP/IP Reference Model [5]. It is introduced here to define the different layer naming in the Internet today.

The main design goal in the ARPANET was the robustness against the drop out of parts of the network. To recover from such failures the focus was on the routing of data packets on the Internet Layer. The Presentation and Session Layers are not present, so the Application Layer and the Transport Layer interact directly.

The reference model for today’s Internet is similar to the TCP/IP model. The differences are on the lower layers. The Internet Layer is referred to as Network Layer. The Link Layer is subdivided into a Data Link and Physical sublayer.
2.2 IEEE 802 Protocol Architecture

The Institute of Electrical and Electronics Engineers (IEEE) provides with the IEEE 802 protocol architecture an implementation of the TCP/IP reference models Link Layer. The standard includes protocols that define a link for transmission of digital data using various mediums. Like copper wire in IEEE 802.3, wireless radio in IEEE 802.11. It also includes protocols to implement message passing between these links to provide a bridge from one medium to another in IEEE 802.1 and 802.2. Here only a briefly introduction to the different sublayer and to the used nomenclature is given.

2.2.1 Architecture and Service

The IEEE 802 architecture standardize the functionality and design of a Local Area Network (LAN) [13]. The function of a LAN is defined in the lower two layers of the OSI Reference model.

For some of the IEEE 802 standards, the data link layer and the physical layer are subdivided into two sublayers, which are in the case of 802.11:

**Logical Link Control (LLC)** controls the transfer of protocol data units (PDU) with clients that share the same physical medium.

**Medium Access Control (MAC)** Receives and sends blocks of data from and to the LLC layer and is responsible for performing functions related to medium access and transmitting the data.

**Physical layer convergence procedure (PLMC)** Defines a method of mapping 802.11 MAC layer protocol data units (MPDUs) into a framing format suitable
CHAPTER 2. BACKGROUND

Figure 2.3: The lower layers from OSI and IEEE 802 model

for sending and receiving user data and management information between two or more stations using the associated PMD sublayer.

Physical medium dependent sublayer (PMD) Defines the methods of transmitting and receiving user data over the physical medium.

The IEEE standard knows three different sorts of addresses. All addresses have a size of 48 bit. Every device has a build-in individual address. Transmissions addressed to the individual address of a stations network interface device is called a unicast transmission. The other addresses are for addressing transmissions to groups of stations. The so-called broadcast address addresses all devices attached to a LAN. The groupcast addresses address a group of devices attached to a LAN. A device forward up a received transmission if either the device is the address unicast receiver or the transmission is addressed to a group or the broadcast address. If a station is not member of a group the group addressed transmission is dropped in the Network Layer.

The group membership management is not defined in IEEE. It is defined in the Network Layer. The IP protocol, introduced in a later section, provide a mechanism to map IP multicast addresses to IEEE groupcast addresses. The membership management is implemented in the Internet Group Management Protocol, also introduced in a later section.

2.2.2 Other IEEE 802 Standards

The IEEE 802 Standard consist of a number of standards. Some of these standards are presented here to provide a small overview over the architecture of IEEE 802.


2.3 IEEE 802.11 Wireless Network

In this section an introduction to the IEEE 802.11 wireless network standard as it is defined in the version from 2007 named 802.11-2007 is given. The standard contains different supplements and amendments, like the different physical layer implementations for 2.4 and 5 GHz frequency band or the Quality of Service (QoS) extensions.

Here the focus is on the MAC sublayer functionality, as far as it is needed for the understanding of the IEEE 802.11aa amendment. In addition we give a short introduction into the physical medium dependent sublayer defined in IEEE 802.11a.

2.3.1 Architecture

There exist two topologies for wireless LAN. The one is Ad Hoc Networking. In this topology there is no central wireless station that connects the wireless stations with each other. The data frames are sent directly from the sender to the receiver. If no transmission between sender and receiver is possible, the data can not be sent. Protocols that use intermediate wireless stations as relay stations to forward such transmissions are called mesh-networks. This mode is referred to as Independent Basic Service Set (IBSS) in the standard.

The other topology is called infrastructure mode or managed mode. In this mode a wireless station connects the other wireless stations with each other. The central wireless station is called access point (AP). Every data frame is sent from or to this access point. This is known as Basic Service Set (BSS). A wireless LAN with multiple access points that are connected via a wired backbone is called an Extended Service Set (ESS). An ESS is used to cover areas larger than the range of a single access point with wireless LAN. The IEEE 802.11 provides a service that ensures that a mobile wireless station always associate with the next access point.
2.3.2 Frame Types

In IEEE 802.11 three different types of frames are defined. The management frames, the control frames and the data frame. For every frame type several subtypes are defined.

The management frames are the frames used for association, authentication and the setup of agreements between stations. Like the Block Ack agreement. The control frames are the Ack, Block Ack Request, Block Ack Response and the RTS/CTS frames. They are used for transmission flow control. The data frame contains the transmitted data. It exist several different subtypes for QoS and HCCA support.

The data is referred to as Mac Service Data Unit (MSDU) in the context of data forwarded to or from the upper layer. In the context of data forwarded to or from the lower layer it is referred to a Mac Protocol Data Unit (MPDU). A MSDU can be fragmented into up to 16 MSDUs.

2.3.3 Medium Access Control

Since the wireless channel is a shared medium only one station can transmit successfully at a time. The different methods given in the IEEE 802.11 Medium Access Control (MAC) have to coordinate this access to the channel.

![Figure 2.4: The MAC architecture in IEEE 802.11. Image: IEEE](image)

The data received from the upper layer is transmitted in frames. Every frame consists of a header, a frame body for the data with variable length and a frame check sequence. The header contains frame control, duration, up to four addresses, sequence control and, for QoS frames, QoS information. The frame control determines the type of the frame. Duration is the time needed to transmit the frame and a possible reply. The addresses denote the intended receiver or receivers of the frame and the source. Sequence control is a sequence number assigned to every data frame. The QoS information contains various information about the frame. Most importantly the priority of the frame, determining the order in which the frames are transmitted.
With the duration and intended receivers provided in the header the wireless stations can implement a protection and power save mechanism. The power saving is done via a method called virtual carrier sensing. If a station receives a header not addressed to it, the station can turn off the wireless radio for the provided duration and save the energy. If the transmission of a frame invokes the transmission of a frame as reply or it is intended to transmit more than one frame in a row, the duration of the first frame can be set to a value that covers the complete transmission sequence. By doing this, other stations are aware of an ongoing transmission and do not interfere.

**DCF**

The fundamental access method of the IEEE 802.11 MAC is a distributed coordination function (DCF), also known as carrier sense multiple access with collision avoidance (CSMA/CA). Each station senses the wireless channel and only starts to transmit if the channel is free. This means the received power on the channel is below a threshold for a given period of time. If it is above this threshold this could mean two things. Either another transmission is using the channel or the channel is too noisy to transmit data successfully.

![DCF Diagram](image)

**Figure 2.5:** The distributed coordination function (DCF). A station is allowed to send if the channel is free for some given time. Image: IEEE

The channel has to be free for a given period between two consecutive frames. This time is called inter-frame space (IFS). Figure 2.6 shows the relations between some IFSs. In DCF the period of time the channel has to be free before a data frame can be sent is a DIFS. If the channel is not free after one DIFS, the station runs a random back-off algorithm. It draws a random number from the so called contention window. Now the station waits one DIFS and for the chosen random number times the slot time before it tries to send again.

The DCF can be used in Ad Hoc and in infrastructure wireless LAN. With DCF a problem known as the Hidden Node Problem arises. If at least two stations want to communicate with a third station and do not receive the transmissions of each other they jam the other stations transmission with their own transmission. The DCF can only work if a wireless station is aware of the other stations transmissions. The solution to this problem is the RTS/CTS mechanism. Before the transmission of
the data frame, the sender transmits a Request To Send frame (RTS). The receiver acknowledges with a Clear To Send frame (CTS). By now a hidden node would have received at least one of the two frames and can hold its own transmission back.

Figure 2.7: Example of the hidden node problem. Two stations are associated with an access point but do not sense the transmissions of each other. Image: IEEE

PCF

The IEEE 802.11 MAC may also incorporate an optional access method on top of the DCF called point coordination function (PCF), which is only usable in infrastructure network configurations. The access point coordinates the channel access here. The period the access to the wireless medium is under control of the PCF is also referred to as contention free period (CFP). The period under control of the DCF is called contention period (CP).

If the wireless channel is idle for one PIFS the access point can poll the stations configured for PCF. A polled station can immediately send one data frame. In the contention free period the channel must only be idle for a SIFS. Figure 2.5 shows that PIFS and SIFS are both shorter than DIFS. An access point can occupy the wireless channel and prevent DCF only stations to get access. To prevent this the duration of the CFP is limited and must be followed by a CP. If the channel is already busy in the CFP, the CFP duration is shorten.
Hybrid coordination function (HCF)

The Hybrid coordination function (HCF) is intended for use in QoS network configurations. The HCF was introduced in IEEE 802.11e and combines functions from the DCF and PCF with some enhanced, QoS-specific mechanisms. The HCF uses both a contention-based channel access method, called the enhanced distributed channel access (EDCA) mechanism for contention-based transfer and a controlled channel access, referred to as the HCF controlled channel access (HCCA) mechanism, for contention-free transfer. The access to the wireless channel under HCF is referred to as a transmission opportunity (TXOP). Each TXOP is defined by a starting time and a maximal duration. A wireless station can win a TXOP in EDCA or poll the access point for a TXOP which grants the TXOP in HCCA.

HCF contention-based channel access (EDCA)

The EDCA is an extension of the DCF. In EDCA four access categories (AC) are defined. The categories are voice, video, best effort and background traffic. Voice traffic has the highest and background traffic the lowest priority. The IEEE 802.1D defines a user priority flag that can be set in the header of each frame. Table 2.2 shows how this user priority is mapped to the EDCA access categories.

<table>
<thead>
<tr>
<th>UP (as in 802.1D)</th>
<th>802.1D designation</th>
<th>AC</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BK</td>
<td>AC_BG</td>
<td>Background</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>AC_BG</td>
<td>Background</td>
</tr>
<tr>
<td>0</td>
<td>BE</td>
<td>AC_BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>3</td>
<td>BE</td>
<td>AC_BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>4</td>
<td>CL</td>
<td>AC_VI</td>
<td>Video</td>
</tr>
<tr>
<td>5</td>
<td>VI</td>
<td>AC_VI</td>
<td>Video</td>
</tr>
<tr>
<td>6</td>
<td>VO</td>
<td>AC_VO</td>
<td>Voice</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
<td>AC_VO</td>
<td>Voice</td>
</tr>
</tbody>
</table>

Table 2.2: Mapping IEEE 802.1D user priority to access categories in EDCA

The maximum duration in ms of a TXOP is announced by the access point in the periodically sent beacon frames. The beacon frame contains an EDCA Parameter Set information element. A limit of 0 indicates that only a single data frame, in
addition to a RTS/CTS exchange and acknowledgment, may be transmitted in a TXOP independent from the used coding rate. A wireless station can transmit frames with different coding rates within a TXOP. This may occur if a data frame is retransmitted in a more robust rate due to a transmission error. It is the responsibility of the station that wins the TXOP to ensure that the transmission and the immediate feedback do not occupy the wireless medium longer than the maximum TXOP duration.

A station obtains a TXOP similar to the access defined for the DCF. For every access category an arbitration IFS (AIFS) is defined along with a maximum size of the contention window. AIFS[AC] denotes the AIFS duration for AC. AIFS is given by the formula

\[ AIFS[AC] = AIFSN[AC] \cdot a\text{SlotTime} + aSIFSTime \]

where AIFSN[AC] is announced by the access point in the EDCA Parameter Set. In table 2.3 the values for the different IFS are given. AIFSN[AC] is set to the default, given in 802.11a.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Duration</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slot time</td>
<td>( T_{SLOT} = 9\mu s )</td>
<td>fixed by physical layer</td>
</tr>
<tr>
<td>SIFS</td>
<td>( T_{SIFS} = 16\mu s )</td>
<td>fixed by physical layer</td>
</tr>
<tr>
<td>DIFS</td>
<td>( T_{DIFS} = 34\mu s )</td>
<td>( 2 \times T_{SLOT} + T_{SIFS} )</td>
</tr>
<tr>
<td>AIFS[VO]</td>
<td>( T_{AIFS[VO]} = 34\mu s )</td>
<td>( 2 \times T_{SLOT} + T_{SIFS} )</td>
</tr>
<tr>
<td>AIFS[VI]</td>
<td>( T_{AIFS[VI]} = 34\mu s )</td>
<td>( 2 \times T_{SLOT} + T_{SIFS} )</td>
</tr>
<tr>
<td>AIFS[BE]</td>
<td>( T_{AIFS[BE]} = 43\mu s )</td>
<td>( 3 \times T_{SLOT} + T_{SIFS} )</td>
</tr>
<tr>
<td>AIFS[BG]</td>
<td>( T_{AIFS[BG]} = 79\mu s )</td>
<td>( 7 \times T_{SLOT} + T_{SIFS} )</td>
</tr>
</tbody>
</table>

Table 2.3: IFS used in EDCA. AIFSN[AC] is set to the default in 802.11a.

If a station senses the wireless channel idle for the IFS duration of the access category it wants to send frames for, it can start to send the frames. If the station does not sense the channel idle, this is most likely due to other stations that want to transmit at the same time. To overcome the problem of more than one station accessing the channel at the same time an exponential back off algorithm is applied. Before the next transmission the station must now sense the channel idle for the appropriate IFS duration and in addition for the duration of a slot time times a random number drawn from the contention window.

Figure 2.9 shows the initial size of the contention window in DCF is 7. The size of the contention window in DCF and EDCA is doubled after each transmission error. With every round the probability that the stations draw different numbers and wait for a different time increase. So does the probability that one station gets access to the channel before the others and can transmit. After a successful transmission the contention window is reset to its minimal size. If the number of retries exceeds the allowed retry limit the frame is dropped and the failure is signaled to the upper layer.
The contention window size in EDCA is defined as $CW[AC]$ for each access category. Initially the contention window is minimal for every access category. If a collision occurs the back off algorithm is used to defer the channel access. In EDCA there are two types of collision. The external collision, like in DCF, when two stations try simultaneously to get channel access. The internal collision, when two access categories in one station try simultaneously to get channel access. In 802.11a the external and internal collisions are handled in the same way. The EDCAF for each access category runs an independent instance of the back off algorithm. Table 2.4 shows how the contention windows are defined for each access category. In the EDCA Parameter Set the ECWmin and ECWmax fields determine the $CW_{min}$ and $CW_{max}$:

$$CW_{min} = 2^{ECW_{min}} - 1$$
$$CW_{max} = 2^{ECW_{max}} - 1$$

The minimum encoded value of $CW_{min}$ and $CW_{max}$ is 0, and the maximum value is 32 767. The default for 802.11a is $CW_{min} = 15$ and $CW_{max} = 1023$.

<table>
<thead>
<tr>
<th>AC</th>
<th>AIFSN</th>
<th>CWmin</th>
<th>CWmax</th>
<th>Default min/max</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td>7</td>
<td>aCWmin</td>
<td>aCWmax</td>
<td>15/1023</td>
</tr>
<tr>
<td>BE</td>
<td>3</td>
<td>aCWmin</td>
<td>aCWmax</td>
<td>15/1023</td>
</tr>
<tr>
<td>VI</td>
<td>2</td>
<td>(aCWmin+1)/2-1</td>
<td>aCWmin</td>
<td>7/15</td>
</tr>
<tr>
<td>VO</td>
<td>2</td>
<td>(aCWmin+1)/4-1</td>
<td>(aCWmin+1)/2-1</td>
<td>3/7</td>
</tr>
</tbody>
</table>

Table 2.4: The contention windows are defined in a way to provide channel access for higher priority access categories in the case of collisions.

With the different AIFS[AC] and the different CW[AC] the access categories voice and video have a higher chance to gain channel access than best effort and background. This is used to transmit data frames of streaming media transmissions with higher priority in the presence of background traffic. Traffic from legacy wireless stations can be transmitted via DCF without interfering the QoS traffic. In the
case of collisions, DCF stands back from the shared medium because of the bigger contention window and the EDCA access categories with higher priority gain access.

**HCF controlled channel access (HCCA)**

The PCF is defined as an alternative to the contention period channel access. The HCCA is an optional extension of the PCF. Since the PIFS used in the PCF and HCCA has shorter duration than the DIFS or AIFS[VO], the HCCA can be used on top of EDCF.

The most important differences compared to the PCF are the following. The TXOP that enables the access point to grant channel access for the transmission of multiple frames at once. The HCCA frame exchange is not limited to the CFP after the beacon. The access point can always start a CFP period by polling for a TXOP in the CP.

### 2.3.4 Acknowledgment and Block Ack Mechanism

In the legacy IEEE 802.11 standard every transmitted data frame is acknowledged with an ACK frame. This is illustrated in figure 2.10.

![Figure 2.10: Every data frame is followed by an ACK frame in unicast transmissions. Image: IEEE](image)

This form of transmission is synchronous transmission. The originator transmits a frame and waits for an acknowledgment. In IEEE 802.11 the originator waits one SIFS. If after one SIFS no Ack frame from the recipient is received by the originator, the originator assumes the frame was not received by the recipient and initiates a retransmission.

With the TXOP the channel usage is more efficient. A station can now transmit several frames without the risk of losing the channel due to the back off algorithm. In addition the time spent in the channel access procedure is saved. In a TXOP
between two consecutive frames must be only one SIFS idle time.

The Block Ack mechanism also provides a more efficient channel usage by aggregating several acknowledgments into one frame. The transmitting station must set up a Block Ack agreement with the receiver, then it can transmit data frames without any acknowledgment by the receiver. After the transmission the transmitter sends a Block Ack Request frame to the receiver and the receiver answers with a Block Ack frame.

This form of transmission is asynchronous transmission. The originator transmits frames without waiting for an acknowledgment and maintains a buffer with the transmitted frames for repetition in case of an error. The recipient maintains a buffer for reordering the received frames in case of retransmissions and a cache to keep track of the already received frames.

The Block Ack mechanism is only available if originator and recipient provide both support for QoS. The support is needed because of the sequence numbering in QoS. Legacy 802.11 defines one sequence number counter per station. With QoS each station maintains a sequence number counter per TID and recipient for QoS data frames. This means that for the management frames, non-QoS data frames and all frames addressed to group or the broadcast address a single sequence number counter is used. The TID is the traffic identifier, a field introduced in 802.11e with the QoS support. It is present in QoS data frames and stores the user priority that define the EDCA access category to use for that frame.

**Block Ack Agreement**

An Block Ack agreement exists between the originator and the recipient for a TID. The set-up of a Block Ack agreement is done via the exchange of Add Block Acknowledgment (ADDBA) frames. The Block Ack agreement can be terminated via the exchange of a delete Block Acknowledgment (DELBA) frame. If an agreement exists, normal acknowledgment or Block Ack can be used.

In the set-up phase the two stations agree on Immediate Block Ack or Delayed Block Ack. In the immediate mode the receiver shall send the Block Ack frame immediately after the reception of the Block Ack Request frame. In Delayed Block Ack the receiver can send the Block Ack Response frame later, but Block Ack and Block Ack Response frame must be acknowledged with standard Ack frames. The delayed mode exists mostly, according to the IEEE 802.11-2007, to enable cheaper or older wireless devices to preform the processing at the host.

Also in the set-up phase the two stations agree on the number of MPDUs that can be transmitted before an Block Ack Request must be sent. This value is limited on the one hand by the available physical buffers in the stations. On the other hand it is limited by the properties of the sliding window protocol that the Block Ack mechanism implements. The standard defines that the buffer size that is announced
in the ADDBA frames is in the range of 0 to 127. The actual buffer size is in the range of 1 to 128 MPDUs. A MPDU is limited to a length of maximal 2304 octets. This gains a maximal buffer size of $288 \text{kBytes}$ for a single agreement in one station.

In addition the recipient needs to store a cache per agreement. The cache stores for each received QoS data frame a tuple with the originators address and the TID. The address has 6 octets and the TID field 2 octets. This results in a maximal cache size of $1 \text{kBytes}$. While the buffer can be implemented in the hosts memory, the cache must be implemented near the device. The Immediate Block Ack defines that the Block Ack is transmitted one SIFS after the end of the Block Ack Request. This means within $16 \mu s$. If the device does not provide enough storage or the host can’t process the data in time, the cache can be implemented in the hosts memory. In this case Delayed Block Ack must be used.

![Message sequence chart for Block Ack mechanism](image)

Figure 2.11: Message sequence chart for Block Ack mechanism. Image: IEEE

**Block Ack Operation**

The originator buffer limits the number of MPDUs that can be send with outstanding acknowledgment. For every outstanding MPDU a copy must be kept for a retransmission in case of an error. The total number of MPDUs in transit shall not exceed the size of the recipient buffer. The recipient must buffer the received MPDUs and must only forward up complete MSDUs ordered by sequence number. In addition the recipient keeps track of the received MPDUs in a separate cache. The cache is needed at the recipients side for the Block Ack generation. The recipient will forward up a received QoS data frames up to the Network Layer as soon as the corresponding MSDU is received completely and no MPDUs from a MSDU with an older sequence number is in the buffer.

The sequence number space is of size $2^{12}$. The space is divided into the parts “new”
and “old”. The “new” range starts with the start of the receiving buffer and ends at the receiving buffers start plus $2^{11}$ modulo $2^{12}$. The rest is the “old” range.

The recipient only buffers received MPDUs when it recognizes a missing frame. This is done via the sequence numbering. Thus in an ideal case the recipients buffer is empty. The recipient generates the Block Ack frame with the information about the received QoS data frames in the cache. The originator should limit its transmissions of MPDUs with outstanding acknowledgment to the recipients reorder buffer and cache size.

The Block Ack Request frame contains a starting sequence number. This number is set to the oldest MPDU with outstanding acknowledgment, which marks the start of the window. The recipient can now acknowledge the reception of up to 64 MSDUs in one Block Ack Response frame.

In the Block Ack the starting sequence number from the request is used as offset for a field of 128 octets. This 1024 bits correspond to the possible $16 \times 64$ MPDUs from up to 64 fragmented MSDUs. The recipient sets for all received MPDUs the corresponding bits to 1 and all other bits to 0. The originator schedules the not acknowledged MPDUs after the reception of the Block Ack for retransmission.

The Block Ack mechanisms is designed to reduce overhead for unicast transmissions. The IEEE enhances in 802.11aa the Block Ack mechanism to support group addressed transmissions. There was no feedback mechanism for multi- or broadcast transmission in 802.11 before. The IEEE 802.11aa, that is still draft in 2009, introduces More Reliable Groupcast (MRG) support. The MRG support is presented in a later section.

### 2.3.5 Radio Propagation and Rayleigh fading Channel

The physical channel is the electromagnetic spectrum. Electromagnetic waves propagate from a transmitter to a receiver. An electromagnetic wave can be described as a periodic signal which again is defined by its wavelength $\lambda$ and frequency $f$. The frequency is the rate at which the signal repeats. The wavelength is the distance the signal propagates in this time. The relationship between $\lambda$ and $f$ is $\lambda f = v$. For electromagnetic waves the velocity $v$ is $v = c$, the speed of light in free space, which is approximately $3 \times 10^8 m/s$. The unit for frequency is expressed as cycles per second, or Hertz. The wavelength is expressed in meters. An equivalent parameter for the frequency is the period $T$, which is the amount of time it takes for one repetition, defined as $T = 1/f$. The phase describes the relative position within one period.

A channel can be described by the properties data rate, bandwidth, noise and error rate. The data rate $C$ is the rate at which data can be transmitted, in bits per second (bps). The bandwidth $B$ is the frequency range used by the channel. Noise $N$ is the electromagnetic radiation that the receiver receives but is not transmitted
by the sender. The error rate is the probability that data sent over the channel is not received correctly. The theoretical maximum data rate that can be achieved over a noisy channel is given by the Shannon-Hartley theorem

\[ C = B \log_2 \left( 1 + \frac{S}{N} \right) \]

Where \( S \) denotes the signal transmitted by the sender. The fraction \( S/N \) is the signal-to-noise ratio. The relationship expresses the amount by which the power of the signal exceeds the noise level. This is expressed in terms of decibels

\[ SNR_{db} = 10 \log_{10} \frac{signalpower}{noisepower} \]

The strength of the signal disperses with distance. This form of attenuation is known as free space loss and occurs due to the fact that the energy is spread over a larger and larger area.

Another problem is fading. If there is a line of sight between sender and receiver the electromagnetic waves may directly move from one to the other. But from the antenna the signal propagates in more than one direction. When the signal hits an obstacle it is reflected, diffracted or scattered. The type of deflection depends on the size of the obstacle and the wavelength. For the frequencies used by wireless LAN an obstacle of the size of a lamp post or a traffic sign can cause scatter. The signal is spread in a beam of several weaker signals. For a large obstacle like a wall the signal can be reflected, and at the edge of a large impenetrable object diffraction can occur. In this case the signal is also spread in a beam of weaker signals. Due to this propagation the signal may reach the receiver on more than one path. Since the propagation speed is constant and the path is variable, the signal will reach the receiver at different points in time and with different power. These copies of the original signal add up at the receiver. They may cancel out or amplify each other.

For digital transmissions this is known as Inter Symbol Interference (ISI). A short pulse, encoding one or more bits of data, is transmitted. The pulse reaches the receiver at different times and power levels. These copies of the original symbol add a kind of noise to the received pulse, making it more difficult to tell it apart from the adjacent symbols. This problem gets worse with higher data rates, when symbols are sent closer in time.

To estimate the effect of signal degradation on the channel, models are used. The Rayleigh fading describes the effect of the multi-path propagation described above. The basic assumption in the model is that no dominant line-of-sight exists and the main amount of the signal power reaches the receiver on different paths. This is most applicable on the environment wireless LAN is used in. A detailed description of this mathematical model can be found in [10].
2.3.6 Modulation and Coding in IEEE 802.11a

The IEEE 802.11 defines multiple physical depend sublayers. 802.11b and g operate in the 2.4 GHz band, with data rates up to 54 Mbps for 802.11g. The 802.11a supports data rates up to 54 Mbps, but operates in the 5 GHz band. We focus on the 802.11a here because the 5 GHz band is not as used by other wireless systems as the 2.4 GHz band and the modulation, used by 802.11a, better compensates for problems like fading.

In 802.11a Orthogonal Frequency Division Multiplexing (OFDM) is applied. OFDM uses multiple carrier signals at different frequencies. A channel with a center frequency $f_b$ and a bandwidth of $Nf_b$, is subdivided into $N$ subcarriers. The lowest subcarrier is $f_b$, all the other subcarrier frequencies are multiples of $f_b$. In the frequency domain the subcarrier are orthogonal to each other. As one can see in Figure 2.12, the peak in the power spectral density of every subcarrier is at the point at which it is zero for the adjacent subcarriers.

![Figure 2.12: Illustration of Orthogonality in OFDM](image-url)
The data is now not sent in serial over the complete channel. Instead it is sent in parallel over the subcarriers. Frequency selective fading only affects some subcarriers, but not the complete channel and so not the complete transmission. More important is that the Inter Symbol Interference is reduced. The data rate for a subcarrier is $N$ times smaller than for the complete channel.

In 802.11a a channel is divided into 52 subcarriers. 48 for data transmission and 4 are used as pilot subcarriers. Pilot subcarriers are used to make the transmission more robust against frequency offset and phase noise. Table 2.5 shows the data rates that can be archived using the different modulations. An OFDM symbol is the pulse transmitted over the channel. In the symbol the bits are encoded to the different subcarriers. In 802.11a an OFDM symbol has a duration of $4\mu s$.

<table>
<thead>
<tr>
<th>Data rate</th>
<th>Modulation</th>
<th>Code rate</th>
<th>Coded bits per OFDM symbol</th>
<th>Data bits per OFDM symbol</th>
<th>Channel holding time (1500 byte)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>BPSK</td>
<td>1/2</td>
<td>48</td>
<td>24</td>
<td>2ms</td>
</tr>
<tr>
<td>9</td>
<td>BPSK</td>
<td>3/4</td>
<td>48</td>
<td>36</td>
<td>1.3ms</td>
</tr>
<tr>
<td>12</td>
<td>4-QAM</td>
<td>1/2</td>
<td>96</td>
<td>48</td>
<td>1ms</td>
</tr>
<tr>
<td>18</td>
<td>4-QAM</td>
<td>3/4</td>
<td>96</td>
<td>72</td>
<td>666µs</td>
</tr>
<tr>
<td>24</td>
<td>16-QAM</td>
<td>1/2</td>
<td>192</td>
<td>96</td>
<td>500µs</td>
</tr>
<tr>
<td>36</td>
<td>16-QAM</td>
<td>3/4</td>
<td>192</td>
<td>144</td>
<td>333µs</td>
</tr>
<tr>
<td>48</td>
<td>64-QAM</td>
<td>2/3</td>
<td>288</td>
<td>192</td>
<td>250µs</td>
</tr>
<tr>
<td>54</td>
<td>64-QAM</td>
<td>3/4</td>
<td>288</td>
<td>216</td>
<td>222µs</td>
</tr>
</tbody>
</table>

Table 2.5: Data rates in IEEE 802.11a with a channel bandwidth of 20 MHz

Table 2.5 shows that 802.11a supports multiple combinations of modulation and coding alternatives to support different channel conditions. A convolutional code, with code rates of 1/2, 2/3 and 3/4, provides forward error correction.

![Figure 2.13: Modulation of analog signal for digital data using BPSK](image)

Figure 2.13 shows an example of Binary Phase-Shift Keying (BPSK). BPSK, also known as two-level PSK, is the simplest PSK scheme. Two phases represent one symbol in this encoding and every symbol one of the two binary digits. Every time a 0 is transmitted, the basic sine wave is shifted by $\pi$.

A more efficient modulation can be achieved if each symbol represents more than one bit. Quadrature Amplitude Modulation (QAM) takes advantage of the fact that one can send two different signals simultaneously on the same frequency. In QAM two copies of the carrier, a basic sine wave, are sent, one shifted by $\pi/2$. Each carrier...
is then modulated via Amplitude-Shift Keying (ASK). In ASK different values are represented by different amplitudes of the carrier. The basic case is to send no carrier to represent a 0 and the carrier to represent a 1. In 4-QAM a two-level ASK is used. Each carrier can represent 2 states and combine \(4 = 2 \times 2\) states. For more ASK levels and smaller fractions more states can be defined, in IEEE 802.11a up to \(64 = 16 \times 16\) states. Of course the robustness against attenuation and noise is reduced, if the number of states, that must be distinguishable, is increased.

### 2.3.7 Other IEEE 802.11 Standards

The IEEE 802.11 Standard consists of a number of standards. Some of these standards are presented here to provide a small overview over the architecture of IEEE 802.11. The supplements a, b, g, h, i and e form together the main part of the new version of the IEEE 802.11 Standard, named IEEE 802.11-2007.

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11</td>
<td>MAC and Physical Layer at 2.4 GHz with 1 and 2 Mbps.</td>
</tr>
<tr>
<td>802.11a</td>
<td>Physical Layer at 5 GHz with up to 54 Mbps.</td>
</tr>
<tr>
<td>802.11b</td>
<td>Physical Layer at 2.4 GHz with up to 11 Mbps.</td>
</tr>
<tr>
<td>802.11g</td>
<td>Physical Layer at 2.4 GHz with up to 54 Mbps.</td>
</tr>
<tr>
<td>802.11h</td>
<td>Extensions for international regulations to physical and MAC layer.</td>
</tr>
<tr>
<td>802.11i</td>
<td>MAC Layer security and authentication: WPA and WPA2.</td>
</tr>
<tr>
<td>802.11e</td>
<td>MAC Layer improvement for QoS: EDCA and HCCA.</td>
</tr>
<tr>
<td>802.11n</td>
<td>MAC and Physical Layer enhancements to enable higher throughput. (Ongoing)</td>
</tr>
<tr>
<td>802.11aa</td>
<td>MAC Layer extension to realize robust streaming of audio/visual transport streams. (Ongoing)</td>
</tr>
</tbody>
</table>

Table 2.6: Some of the IEEE 802.11 Standards

### 2.4 Internet Protocol

In this section parts of the Network Layer are introduced. The focus is on parts that are interesting for the understanding of the functionality between Link Layer and the Network Layer with respect to the IEEE 802.11aa amendment.

#### 2.4.1 Overview

The Internet Protocol (IP) is used to send data packets across packet switched networks. The basic part is defined in RFC 791 [25]. The protocol is used in the OSI Network Protocol Layer.
The nodes of an IP network can be connected via one or more IEEE 802 based networks or another network that provides the same functionality to the OSI Network Protocol Layer. In this document only some of the extensions to the IP that are used to provide QoS and support for multicast traffic are mentioned.

2.4.2 Differentiated Services

Differentiated Services (DiffServ) is a method to implement QoS in IP networks. It is defined in RFCs 2474 [21] and 2475 [4]. The header of every IP packet contains an Type of Service (ToS) field. This field is redefined in DiffServ to define groups with different QoS behavior. To make it distinguishable from the ToS field, it is referred to as DiffServ Code Point (DSCP). An IP Packet can belong to one out of 64 groups. The QoS is implemented via the behavior of each IP node the packet visits. The node has a set of predefined Service Level Agreements. They define the priority of packets from a group at this node.

Since it depends on the trust every node needs to have in the DSCPs and the behavior of the other nodes, it is only for use in IP networks under a single administration or multiple administrations that trust each other. An example is the IP network of an Internet Service Provider (ISP).

2.4.3 Multicast Transmission

If the number of people listening to an analog radio broadcast increases, the radio station does not need to transmit more data. This does not hold true for IP based networks. Applications like IP based audio/video transmissions cause more traffic on the network if the number of listeners increases. A method to overcome this problem is multicast transmission. The IP multicast is defined in RFC 1112 [5].

The idea behind multicast is that a sender transmits a packet only once. This packet is addressed to a group of stations, the so-called multicast address. The network forwards the packet to its receivers and duplicates it as late as possible to avoid multiple transmissions. To provide this at every node in the IP network, the routers, a multicast routing protocol must be run.

Multicast Addressing

The IP address range from 224.0.0.0 to 239.255.255.255 is the IP multicast address range. This range is also referred to as Class D. The IP multicast address assignment is defined in RFC 3171 [3]. In addition to this exists a broadcast address 255.255.255.255 for packets that shall be received by every station on a network.

The simplest implementation of IP multicast transmission after the last router is to send the multicast packet to the broadcast address. This does not cause any
additional congestion if all stations of this network are in one collision domain.

For the IEEE 802 LAN a special mapping from the IP multicast to the IEEE 802 groupcast addressing is defined. This is of special interest for multicast transmission on IEEE 802.11 wireless LAN. In IEEE 802.11 a wireless station can turn off its radio and save power if a transmitted frames is of no interest to the station. Some switches for IEEE 802.3 networks inspect the content of IGMP messages to learn the collision domains the groupcast frames must be forward to. This is called IGMP-snooping, RFC 4541 [7].

The mapping of an IP multicast address to an IEEE 802 MAC group address is defined as follows. The IEEE MAC address range is from 01:00:05:00:00:00 to 01:00:05:7f:ff:ff is used for group addresses. The lower 23 bit of the 28 bit used in the multicast addressed are mapped to the group address, as illustrated in figure 2.14. This introduce ambiguity in the group addresses. If two stations associated to an access point subscribe two different multicast groups whose addresses only differ in the first 5 bits, the stations can not distinguish between the transmissions and receive both. The non required packets are removed in the hosts IP stack, in this case.

**Internet Group Management Protocol (IGMP)**

The Internet Group Management Protocol (IGMP) is used to manage the membership of multicast groups. It is defined in RFC 1112 [8], Version 2 and 3 are defined in RFC 2236 [9] and RFC 3376 [6].

The IGMP defines three types of messages. The Membership Query, the Membership Report and the Leave Group message. A station joins a multicast group by sending a Membership Report with the corresponding multicast address. The router periodically sends Membership Query messages. If the stations do not reply to this messages in time, the router assumes the station to be gone. The Leave Group message shall be sent by a station that leaves a group, in order to enable the router to update its routing state faster.
Chapter 3

The IEEE 802.11aa Amendment

To overcome the lack of an efficient and more robust support for streaming of audio/visual content to multiple destinations, the Project IEEE 802.11aa Task Group¹ has been formed. The group is working on extensions to the given IEEE 802.11 standard to enable Quality of Service for multicast traffic. The draft amendment describes mechanisms to provide feedback and admission control for multicast traffic, the most important is the so-called More Reliable Groupcast (MRG). The MRG is a feedback mechanism that can be used to determine if a data frame has been lost and to perform link adaptation. In this chapter an introduction to the More Reliable Groupcast is given.

3.1 More Reliable Groupcast Overview

The access point negotiates with its associated stations the MRG policy. It can provide for every group different policies. These MRG policy agreements can be changed or canceled. An agreement exists between an access point and a non-AP station for a group addressed stream. In 802.11aa four acknowledgment policies are defined. No-Ack/No-Retry, MRG-Directed, MRG- Unsolicited-Retry and MRG-Block-Ack.

The No-Ack/No-Retry policy is the fallback default. In this policy the data frames are transmitted without any feedback to the multicast address. This policy is used as long as no other agreement for another policy exists.

In the MRG-Directed policy the group addressed data frames are not transmitted as given. Instead they are transmitted addressed to the unicast address of the non-AP station. This is the most reliable policy. The data frames addressed to a single station are transmitted using the normal unicast acknowledgment policy, so the given link adaptation mechanisms can be used. It is inefficient as soon as more than one station is requesting a group addressed stream, because of the multiple

¹http://www.ieee802.org/11/Reports/tgaa_update.htm
transmission of the same data, once addressed to the group and once addressed to
the station. In the chapter on link adaptation common link adaptation mechanisms
are introduced.

The MRG-Unsolicited-Retry is alike to the No-Ack/No-Retry policy. No feedback is
provided here. To provide more robust streaming, the access point may retransmit
data frames. The retransmission scheme is not part of the standard.

The MRG-Block-Ack policy is the extension of the given unicast Block Ack in
802.11, that was introduced in the background chapter. It extends the sliding
window protocol defined in Block Ack for the use with group addressed traffic.
The MRG Block Ack Request is addressed to the group and contains a list of
the stations that shall reply. A station addressed in the request replies with an
MRG Block Ack Response frame. The MRG Block Ack response frame itself and
the contained information about the received or lost data frames can be used to
provide retransmissions and link adaptation.

The draft does not define how an access point learns about the group membership of
the stations associated to it. It proposes the use of IGMP snooping. IGMP snoop-
ing is a method used in switches to reduce the network load by forwarding group
addressed frames only to ports with networks attached that contain subscribers
to that group. To gain the necessary knowledge about the group membership, the
switch snoops into the higher layer Internet Protocol packets. A definition of IGMP
snooping can be found in [7].

In addition to this the amendment specifies two power management modes that can
be used for group addressed frames. These are All-Active/Any-PS and MRG-SP.
MRG-SP transmits group addressed frames at scheduled Service Periods.

3.2 More Reliable Groupcast Agreement Policy Setup

The setup, change or tear down of an agreement between an access point and an
associated station can be done via the association management frames or via the
new management action frame types MRG Request and MRG Response.

The management frames, like the beacon frame, can contain a variable number of
frame body components. Some are so called Information Elements. They have a
common general form consisting of an Element Id field and a Length field. The
element id is defined in the IEEE 802.11 to name the type of information. For
example the EDCA Parameter Set Information Element has id 12 and a fixed length
of 20 octets. For MRG the ids are not defined yet.

The amendment defines the MRG Request and MRG Response elements. The
request element contains the group address the agreement is for, the MRG Ack
Policy, the MRG Power Management Mode. The optional Schedule Element is
present, if the power management is set to MRG-SP. It is sent by a non-AP station
3.2. MORE RELIABLE GROUPCAST AGREEMENT POLICY SETUP

The request element format is shown in Figure 3.1. The request elements length field is set to either 65 or 79 octets.

<table>
<thead>
<tr>
<th>Element ID</th>
<th>Length</th>
<th>Group Address</th>
<th>MRG Ack Policy</th>
<th>MRG Power Management Mode</th>
<th>TSPEC Element</th>
<th>Schedule element (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>57</td>
<td>0 or 14</td>
</tr>
</tbody>
</table>

Figure 3.1: The MRG Request element format. Field sizes are given in octets.

The response element is sent by the access point to the associated station. It contains the group address the agreement is for and the MRG Ack Policy. If the MRG Policy is denied by the access point, no further fields are present. If not, the MRG Power Management Mode field is present. If the power management mode is MRG-SP, the Schedule Element is also present. The response elements length field is set to 7, 8 or 22 octets, depending on the presents of the optional fields.

<table>
<thead>
<tr>
<th>Element ID</th>
<th>Length</th>
<th>Group Address</th>
<th>MRG Ack Policy</th>
<th>MRG Power Management Mode</th>
<th>Schedule element (optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0 or 14</td>
</tr>
</tbody>
</table>

Figure 3.2: The MRG Response element format. Field sizes are given in octets.

The TSPEC Element is part of the transport stream setup mechanism in IEEE 802.11 and used to transmit parameters for admission control. A non-AP station specifies its traffic flow requirements (data rate, delay bounds, packet size, and others) and requests the access point to create a TSPEC by sending the ADDTS (add TSPEC) management action frame. The access point calculates the existing load based on the current set of issued TSPECs. Based on the current conditions, the access point may accept or deny the new TSPEC request. The transport stream mechanism is not mandatory for MRG Block Ack, but a station may use this to signal preferences to the scheduler. This admission control is explained in greater detail in the chapter on admission control.

The non-AP station may request the setup of an MRG agreement by sending an MRG Request frame or an (re-)association request frame with one or more MRG Request elements.

The Category field is set to Robust AV Streaming for the management action frames. The Action field is 0 for a MRG Request and 1 for a MRG Response frame. The non-AP station chooses a non-zero value for the Dialog Token field and the access point responses with this value. The MRG Request frame may contain zero or more MRG Request Elements. The MRG Response frame may contain one or more MRG Response Elements. The access point can send unsolicited MRG Response frames.
to individual non-AP stations, with Dialog Token field set to zero, to advertise MRG support to non-AP stations. This is also used to indicate the change or cancel of supported MRG policies.

If either the access point or the non-access point station does not implement MRG support, the MRG Request and Response Elements are ignored since their id is unknown.

### 3.3 More Reliable Groupcast Block Acknowledgment

#### 3.3.1 MRG Block Ack Setup

Once an access point has a MRG-Block-Ack agreement with a non-AP station, it shall initiate the setup of an MRG Block Ack agreement.

The amendment defines new ADDBA frames, called Extended ADDBA. The action frames are alike the ones for the unicast Block Ack. The difference is the presence of an Extended Block Ack Parameter Set and an optional group address. In the DELBA frame is only extended by a flag to indicate the optional presence of the group address for MRG Block Ack agreement tear down. In the Extended Block Ack Parameter Set only a flag is defined that indicates the presence of a group address. This flag must be always set for the Extended ADDBA Request and Response frames. The new frame types are necessary for downwards compatibility with legacy Block Ack. Given this construction, a legacy station still can use Block Ack.

The more important differences are that delayed Block Ack is not supported in MRG Block Ack and that in the MRG Block the access point is always the originator and
the non-AP stations are the recipients. This is due to the principle that in BSS only the access point is allowed to broadcast to multiple receivers. Non-AP stations address their transmissions to the access point.

A recipient reassemble the MSDUs from the received MPDUs and forward complete MSDUs ordered by sequence numbers up to the higher layer. In addition to this, the recipient maintains a cache. In this cache the sequence numbers of the last received MPDUs are stored. The sizes of the buffer and cache are the same as for Block Ack. The buffer size limits the number of MPDUs that can be sent unacknowledged. The cache must be big enough to keep track of all of these MPDUs. If for instance all the MSDUs are received correctly, they are all forwarded up.

In IEEE 802.11aa also contains some other extensions to the Block Ack mechanisms from the Project IEEE 802.11n Task Group. The Block Ack mechanism distinguishes between Basic Block Ack and Compressed Block Ack. The first supports fragmentation of a MSDU into up to 16 MPDUs and individual acknowledgment for every single MPDU. This is the legacy Block Ack as defined in IEEE 802.11-2007. The latter has no support for fragmentation but is much smaller in size.

The standard defines that the buffer size that is announced in the ADDBA frames is in the range of 0 to 127. The actual buffer size is in the range of 1 to 128 MPDUs. A MPDU is limited to a length of maximal 2304 octets. This results in a maximal buffer size of $288k$Bytes for a single agreement in one station. In addition the recipient needs to store the cache. This limits the number of the QoS data frame with outstanding acknowledgment in terms of MSDUs to 128 MSDUs for Compressed Block Ack and to 8 MSDUs for Basic Block Ack. This because of the fact that for the fragmentation support space for 16 MPDUs per MSDU must be provided. If A-MSDUs are used, which can be up to 7935 octets long, the needed storage is about $991k$Bytes. An A-MSDU consists of multiple MSDUs aggregated into on MPDU. The A-MPDU is defined by the Project IEEE 802.11n Task Group, and the ADDBA frames already support signaling the additional storage need.

Another extension from the IEEE 802.11n is the support for Multi TID Block Ack. This variant supports the acknowledgment of traffic for multiple TIDs in one Block Ack.

### 3.3.2 MRG Block Ack Operation

The originator may send a MRG Block Ack Request at any time. The starting sequence number, sent in the request, indicates the oldest MPDU with outstanding acknowledgment. If a recipient receives a MRG Block Ack Request, it can forward up all complete received MSDUs with a sequence number older than the starting sequence number.

For MRG Block Ack the request frame is sent to the group address. The BAR Control field is defined as the BA Control field, shown in figure 3.9 except the
MRG flag is not defined in the BAR Control field. The flag Compressed Bitmap toggles the use of the Compressed or the Basic Block Ack variant. For the MRG Block Ack only the use of the Compressed is useful since the group addressed traffic is not subject to fragmentation. Thus the support for fragmentation in the Basic Block Ack variant is not needed.

The MRG Block Ack Request frame contains a MRG BAR Information field. In this field the stations are encoded that shall reply to the MRG Block Ack Request.

Figure 3.6: The MRG BAR Information field format. Field sizes are given in octets.

A recipient is addressed by its association identifier AID. The AID is given to the non-AP stations by the access point in the association. If a recipient is addressed its AID is encoded in a bitmap. This bitmap contains up to 2008 bits and is organized into up to 253 octets. An AID corresponds to bit number \((AID \mod 8)\) in octet number \(\text{floor}(AID/8)\). The MRG BAR Bitmap Control is set to the number of the first octet in the bitmap with a bit set to 1. The MRG BAR Partial Bitmap starts with this octet and ends with the last octet with a bit set to 1. The MRG BAR Information Length is set to the length of the virtual bitmap plus 2 to express the length of the complete MRG BAR Information field.

The recipient that is addressed in an MRG Block Ack Request computes the point in time to transmit the MRG Block Ack after the reception following the formula:

\[(N + 1) \times SIFS + N \times TXTIME(MRGBlockAck)\]

where \(N\) is the position the recipients AID has in the list of addressed AID. The smallest AID addressed is at position \(N = 0\). \(TXTIME\) determines the transmission time of the MRG Block Ack frame with the given data rate. Figure 3.7 illustrates a typical frame exchange using the MRG Block Ack.

The recipients shall use the same data rate for the transmission of the MRG Block Ack frame that was used for the transmission of the MRG Block Ack Request.
frame. The reason is that using a less robust data rate with a higher data rate gains only a higher probability of losing the MRG Block Ack. A more robust data rate would take more time to transmit, most likely colliding with the MRG Block Ack transmission of another recipient.

In the MRG Block Ack frame the addressed recipient sets the MRG flag in the BA Control field to indicate the presence of an MRG Group Address field. In this field the recipient stores the address of the group the feedback is for. The MRG Block Ack frame is addressed to the originator. In the BA Information field the recipient indicates the reception of lost for up to 64 MSDUs.

The BA Information takes either 130 or 10 octets in size. The first two octets is always the starting sequence control. In the Basic Block Ack variant the following 128 octets represent the $16 \times 64$ MPDUs the 64 MSDUs can be fragmented into. The starting sequence control corresponds to the first bit, all other bits to the corresponding following MPDUs sequence numbers. The other variant is called Compressed Block Ack. In this variant a bit represents a MSDU, taking 8 octets in complete. It is only suitable if no fragmentation takes place.

If the MRG Block Ack requests feedback for a MPDU older than the oldest entry in the recipients cache, the recipient reports the MPDU as received. In the Basic Block Ack variant the recipient acknowledges only the MPDUs starting from the starting sequence control until the MPDU with the highest sequence number that has been received.

If the MRG Block Ack Request or one of the MRG Block Ack frames is lost, the originator can send another MRG Block Ack Request. This request can ask
for acknowledgment for the same sequence numbers. But the originator shall not address in this request a recipient that successfully sent a MRG Block Ack for the original request. The originator may repeat this until it got a response from all addressed stations or the lifetime limit of the MPDUs, acknowledgment is requested for, is reached.
Chapter 4

Link Adaptation

4.1 Overview

The IEEE 802.11 standard defines more than one physical data rate to transmit data over the wireless medium. The algorithm how to adapt this encoding to changing conditions of the wireless channel is not part of the standard.

The link adaptation algorithm has a strong influence on the reliability of the data transmission over the wireless link. The algorithm that adapts better and faster to the changing wireless channel introduces less delay and overhead. The delay is produced by retransmissions needed because of lost transmissions. The overhead is the redundancy introduced by a more robust rate that occupies the wireless channel for a longer period of time. This time is also referred to as channel holding time (CHT). The physical data rates defined in the IEEE 802.11a are introduced in the background chapter.

The wireless channel is a shared medium. This means only one station can transmit at a time. If two stations transmit at the same time, the transmissions can disturb each other causing both transmissions to be lost. In the IEEE 802.11 standard the Medium Access Control (MAC) controls the access to the channel to prevent this. If a station can release the channel earlier by using a faster, less robust rate, more time for other transmissions is left over.

The simplest approach would be to transmit all frames with the slowest and most robust rate. But this would be very inefficient. By not making the link adaptation part of the standard, the IEEE left room for improvements. The majority of the published algorithms use the individual acknowledgment for every transmitted data frame to measure the throughput. Using this measurements a heuristic decides which encoding is used for the next data frame to be sent. Since all these algorithms depend on a feedback mechanism to provide information about the success of a transmission and the lack of such a mechanism for multi- and broadcast transmission, none of the existing algorithms support the adaptation for this kind of
Some prominent algorithms are presented here. The first group consists of Auto Rate Fallback (ARF), Adaptive Auto Rate Fallback (AARF) and Adaptive Multi Rate Retry (AMRR). They are widely implemented in IEEE 802.11 devices and represent the first generation of algorithms. The next group consists of the Onoe algorithm and the Minstrel algorithm. These two algorithms are implemented in open source wireless device drivers. The Minstrel algorithm is the default algorithm used by the current Linux IEEE 802.11 implementation. All these algorithms have in common that they collect data and adapt the physical data according to a statistical success rate.

In the third group the enhanced versions of the first generation algorithms, mentioned in the first group are presented. The Collision-Aware Rate Adaption (CARA), AARF with Collision Detection (AARF-CD) and the Robust Rate Adaption Algorithm (RRAA) implement heuristics to distinguish between a lost transmission due to a collision with the transmission of a hidden station or due to a weak channel. Based on this information they use the RTS/CTS exchange to protect transmissions or to adapt the physical data rate.

The last group consists of algorithms that use SNR measurement to determine the optimal physical data rate for the next transmission. The first representative is the Receiver Based Auto Rate (RBAR) algorithm. The RBAR algorithm needs changes in the frames transmitted and is not compatible to the IEEE 802.11 standard. The second algorithm is the SNR-based Auto Rate for Multicast (SARM). This algorithm needs additional frames defined for the transmission of the measured SNR values and is specifically designed for the transmission of group addressed audio/visual streaming applications. Since the standard also does not require that a wireless device provides SNR measurement, link adaption based on the measured SNR is upcoming only a short time ago with the availability of better wireless devices.

The ns-3 network simulator implements the link adaption algorithms presented in this chapter, except the SARM algorithm. Some implementations in the ns-3 have small differences compared to the original papers, due to restrictions of the simulation environment.

### 4.2 Auto Rate Fallback

The Auto Rate Fallback (ARF) algorithm from A. Kamerman and L. Monteban [2] was the first published rate adaptation algorithm. It has been published in 1997, the year the IEEE published its first 802.11 standard. The algorithm works as follows:

- If 2 consecutive data frames are not transmitted successfully, send the next data frame with a slower data rate.
4.3. **ADAPTIVE AUTO RATE FALBACK**

- If the first transmission after an increase of the data rate fails, send the next data frame with a slower data rate.
- If 10 consecutive data frames are transmitted successfully or a timeout occurs, send the next data frame with a faster data rate.
- If a data frame is not transmitted successfully, set a timer.

The idea behind the algorithm is that if a series of data frames is transmitted successfully, the channel condition may also allow to transmit successfully at a faster data rate. The first data frame after the increase of the data rate is some sort of probing transmission. If it fails, the algorithm switches back to a more robust data rate rather than waiting for a transmission time times a threshold of packets.

This algorithm suffers from two main problems. The first is the slow convergence to fast changing channel conditions. Since ARF needs 1 or 2 lost and 10 transmitted data frames to switch the used encoding, it might never synchronize to changes of the channel conditions that last shorter than the transmission time of one data frame with a given encoding. Given a channel that does not change over time which does not allow the fastest data rate, the next problem of ARF is that it sends every 10 data frames a probing packet at the next faster encoding. This transmission will fail, due to the constant channel conditions. The result is a constant loss rate on constant channels with ARF.

### 4.3 Adaptive Auto Rate Fallback

To overcome the problems with ARF, in 2004 M. Lacage et al published [16] the Adaptive Auto Rate Fallback (AARF) algorithm. The idea behind this algorithm is to add channel memory. This is based on the assumption that the wireless channel stays comparatively constant for most of the time and the transmission errors occur in bursts and not equally distributed. An example for this model is a user sitting in front of his notebook. Neither the notebook or the stationary access point moves over time.

The channel memory is implemented via a variable threshold for the increase and decrease of the data rate. The period of time between consecutive attempts to use a faster data rate is increased, if an attempt has failed. The algorithm is an extension of the ARF and works as follows:

- If the first transmission after an increase of the data rate fails, send the next data frame with a slower data rate and double the threshold.
- If 2 consecutive data frames are not transmitted successfully, send the next data frame with a slower data rate and reset the threshold to 10.
- If threshold consecutive data frames are transmitted successfully or a timeout occurs, send the next data frame with a faster data rate.
• If a data frame is not transmitted successfully, set a timer.

Lacage limited the threshold of continuously successfully transmitted data frames to 50. AARF is not able to adapt to fast changing channel conditions like the ARF but it does not suffer from a constant loss rate due to probing packages.

4.4 Adaptive Multi Rate Retry

The Adaptive Multi Rate Retry (AMRR) algorithm was presented together with the AARF algorithm by M. Lacage et al [16] in 2004. It is designed to provide a lower latency than the AARF when a transmission fails.

If a transmission fails, the first retry is with the next more robust physical data rate. If the first retry fails, the second retry is with next more robust rate. If the second retry fails, the most robust basic rate is used for the third and last retry. The limit of four retries is due to the history of the hardware it was designed for. The original implementation exploited the possibility of wireless devices based on the AR5212 chip produced by Atheros. This chip provides the feature to transmit a data frame once from the host to the device and queue it for multiple retries at different physical data rates. Due to the properties of this chip the AMRR algorithm was limited to four retries.

The physical data rate for the first try is determined using a heuristic that depends on counting the number of transmissions in the different retry rounds. Every transmission is counted at least in the first round. The heuristic defines a success, a failure and an enough event. The enough event is given when the number of transmissions that was counted in the first round exceeded 10. The condition for the success event is fulfilled when the number of transmissions counted in the first round is 10 times greater than the number of counted transmissions in the second round. A failure is given when the number of counted transmissions in the first round is 3 times smaller than the number of transmissions counted in the second round. The counter are reset to 0 if the enough event is true.

If the enough and the success events are true, the success event counter is increased. When this success counter reaches a threshold the physical data rate is set to the next faster rate and a recovery flag is set. If the failure event it true and the recovery flag is set, the physical data rate is set to the next slower rate and the success threshold is doubled. If only the failure event is true, the physical data rate is set to the next slower rate and the success threshold is reset to its minimal value. The success threshold is in the range of 1 to 10. The exponential growing success threshold models a channel memory to prevent the algorithm from switching to a faster physical data rate to early and to often in order to prevent packet loss. This is a similar approach as seen in the AARF algorithm before.
4.5 Onoe

The Onoe algorithm is named after its author A. Onoe. This was written for the Madwifi Linux kernel module. This module is the device driver for wireless devices based on the AR5212 chip. Thus it is very similar to the AMRR algorithm presented in the previous section. The source code can be found in the repository at the project homepage \[18\]. As the AMRR it tries multiple times to transmit a pending data packet and counts the needed retry rounds. But the heuristic used to determine the physical data rate for the next transmission is different.

The Onoe algorithm tries 4 times to transmit with the current optimal physical data rate. In the next retry round it tries 2 times with the next slower data rate. In the second retry round it tries 2 times with the next slower rate and in the final round 2 times with the most robust physical data rate. The number of successful transmissions, transmission that needed retries and failed transmissions is counted. At a fixed interval of 1000\(ms\) the counters are evaluated and reset to 0. If less than 10% of the transmissions need a retry to succeed in the interval and no one failed, the algorithm increases the credit counter. In all other cases it decreases this counter. If the counter reaches a value of 10, the next faster rate is selected. If the counter reaches a value of \(-10\), the next slower rate is selected. In addition to this the algorithm marks a physical data rate that collected so much discredit and does not select it again for the next 10\(seconds\). This makes the Onoe algorithm a very conservative algorithm without the possibility to adapt to fast changing channel conditions.

4.6 Minstrel

The minstrel algorithm is the current default link adaption algorithm used in Linux. The name is a reference to a medieval minstrel traveling around and staying as long as he gets paid for his endeavor. The algorithm tries at random different physical data rates and stays at the rate which yields the biggest success rate. The authors provide a description of the algorithm at \[20\].

As the other two presented algorithms Onoe and AMRR which were originally designed exploiting the special properties provided by the hardware used, the minstrel algorithm also uses the feature of queuing a transmission for multiple retries at different physical data rates provided by some chips. Because of this the minstrel algorithm also implements up to four tries to send a transmission.

The algorithm uses a fixed percentage of transmissions as so-called look-around transmissions to collect information beyond the current used physical data rate. The default is to use the normal retry scheme for 90% of the transmissions and use 10% of the transmissions as look-around transmissions. For the normal transmissions a fixed decreasing chain of physical data rates is defined. For the look-around
transmissions the chain is selected depending on the randomly selected rate. If the random rate is slower and more robust than the currently used rate, in the first try the current rate is used. This ensures that in a scenario with a channel, that allows transmissions with the fastest physical data rate, the first try is used to test if this rate is still possible and no channel holding time (CHT) is wasted, using a slower rate. If this transmission fails, the next retry rounds are used to collect data on the success probability at different rates. This retry chain is illustrated in figure 4.1.

The physical rate that provides the best throughput is defined as the rate that will provide the highest throughput based on the sampled data. This probable throughput is defined as

$$throughput = \frac{P_{\text{success},i} \cdot 10^6 \text{bit}}{\text{CHT for 1 try at rate i}}.$$  

The probable throughput is computed for every physical data rate $i$. The algorithm updates the success probability for every rate at a constant frequency every 100ms. The influence of new measurements on the selected rate is limited to make the algorithm more robust against outliers. This is done using an exponential weighted moving average that smooths the computed result. The new probability $P_{\text{new},i}$ for a physical data rate $i$ is computed, using the old probability $P_{\text{old},i}$ as

$$P_{\text{new},i} = \frac{(100 - \alpha)P_{\text{sample},i} + \alpha P_{\text{old},i}}{100}$$

where default for weight $\alpha$ is 75 and the success probability in the sample interval $P_{\text{sample},i}$ is defined as

$$P_{\text{sample},i} = \frac{\text{number of successful transmissions}}{\text{number of failed transmissions}}.$$ 

If rate $i$ was not used for transmissions in the sampling interval, no computation is carried out and $P_{\text{success},i}$ remains unchanged. The sampling interval and the smoothing weight are selected to provide a compromise between collecting enough data to provide a meaningful statistic and to recover fast from a bad physical rate selection.
4.7 Collision-Aware Rate Adaption

The Collision-Aware Rate Adaption (CARA) algorithm was published by J. Kim et al in 2007 [14]. The algorithm specifies two methods to differentiate collisions from channel errors to make the rate adaptation only dependent on the channel condition. The key idea is that the transmitting station combines adaptively the RTS/CTS exchange with the clear channel assessment (CCA) functionality to differentiate frame collisions from frame transmission failures caused by channel errors.

The RTS/CTS exchange is based on the assumption that the error probability of a RTS or CTS control frame is negligible, because of the small size and robust transmission rate. Because of this the failure of a RTS/CTS exchange is most probably due to a frame collision. The CARA algorithm enables a so-called RTS/CTS probing if a data frame exchange fails.

The probing is illustrated in figure 4.2. The figure shows the state diagram at a transmitting station. The variables \( m \) and \( n \) are the consecutive success and failure counters. In the **Wait for MPDU** state new data is coming from the upper layer or the current frame transmission fails and a retransmission is requested. In the **DATA Tx** state the frame transmission is completed and the station is waiting for the corresponding Ack frame. In the **RTS Tx** state the station is waiting for a CTS frame after the transmission of a RTS frame. When the number consecutive successes or failures exceed the corresponding thresholds \( M_{th} \) or \( N_{th} \) the physical data rate \( r_{dt} \) is increased or decreased. The default values are \( M_{th} = 10 \) and \( N_{th} = 2 \), providing a behavior like the ARF algorithm. If the size of the pending data frame exceeds a predefined RTS/CTS activation threshold \( RTSThr \) or the number of consecutive failures reaches the RTS/CTS probe activation threshold...
$P_{th}$, a RTS/CTS exchange is initiated by the station before the transmission of the data frame. The default $P_{th}$ is 1, thus on every transmission failure the CARA algorithm first retries the transmission with an RTS/CTS as protection against a collision with a hidden station, and the physical rate is reduced if the retransmission fails again.

The second method to differentiate collisions from channel errors is to use the clear channel assessment (CCA). The CCA is a part of the carrier sensing defined in the IEEE 802.11 standard and describes the procedure to determine if the channel is free or if it is occupied by an ongoing transmission. The so-called CCA Detection in the CARA algorithm is an optional extension and is based on the assumption that in the case of a collision the other transmission is still going on after the end of the data transmission. If the station senses the wireless channel busy after the end of its transmission but does not receive an Ack frame, it is most probable that the transmission was lost due to a collision. This is illustrated in case 2 in figure 4.3. In this case the station will retransmit the data frame without increasing the failure counter or changing the physical data rate. CCA Detection only works in the case the stations are close enough to sense each others transmission and the CSMA/CA can not prevent a collision.

4.8 AARF with Collision Detection

The Adaptive Auto Rate Fallback with Collision Detection (AARF-CD) was published by F. Maguolo et al in 2008 [19]. The algorithm uses an approach like the CARA algorithm. The main differences are the use of adaptive thresholds as in the AARF algorithm and absence of the CCA Detection approach. The authors state that the use of adaptive thresholds provides more stability than the CCA Detection in the presence of hidden stations.

The algorithm is designed as the AARF algorithm, and extended with a RTS/CTS probing mechanism like the CARA algorithm. Unlike the CARA algorithm that turns the RTS/CTS probing off after switching to a faster physical data rate, the AARF-CD algorithm always initiate a RTS/CTS exchange when the physical data
rate has increased. The different behavior provides a higher probability that a possible failure of the succeeding data transmission is due to the channel conditions and an insufficient robust physical data rate and not to a collision with another transmission.

The RTS/CTS probing is controlled via a counter \( rtsCounter \). A RTS frame is transmitted before a data frame if the counter is greater than 0. If a CTS frame is received the counter is decreased by one. The counter is controlled via a RTS window. This RTS window is a variable in the range of 1 to 40. If the physical data rate is increased, the RTS window is reset to 1, turning the RTS/CTS probing on. If the physical data rate is decreased, the RTS/CTS probing is turned off by setting \( rtsCounter = 0 \). If a data frame is lost and the rate was not modified before, the RTS window is doubled if no transmission was successful since the last physical rate change. Otherwise the RTS window is reset to 1. This approach adapt the number of consecutive data frames that are transmitted with preceding RTS/CTS exchange to the assumed collision probability due to hidden stations.

4.9 Robust Rate Adaption Algorithm

The Robust Rate Adaption Algorithm (RRAA) was published by Starsky H. Y. Wong et al in 2006 [29]. The algorithm consists of two sub-algorithm that are combined. The first RRAA-BASIC implements the link adaptation based on the success rate of transmitted data frame. The second is Adaptive RTS (A-RTS). A-RTS uses the RTS/CTS frame exchange to make the link adaptation more robust in the presence of hidden stations. The testbed implementation described in the publication provides retransmission chains as described for the AMRR and other algorithms presented in this chapter. RRAA is implemented on top of this retry mechanism. The RRAA-BASIC provides the physical data rate for the first transmission attempt.

The basic RRAA defines a Maximum Tolerable Loss threshold (MTL) and an Opportunistic Rate Increase threshold (ORI). The algorithm starts to transmit with the fastest supported physical rate and count the number of transmitted and successful data frames. After the transmission of estimation window (ewnd) many data frames the algorithm computes the loss ratio as

\[
P = \frac{\text{lost frames}}{\text{transmitted frames}}.
\]

The physical data rate is decreased to the next more robust rate if \( P \) is larger than \( P_{MTL} \). It is increased if \( P \) is smaller than \( P_{ORI} \). The variables are defined per physical data rate. Figure 4.4 presents the values for the physical data rates defined in IEEE 802.11a. The precise parameter derivation can be found in the publication. It is based on a heuristic that takes the different channel holding time for different
physical data rates into account to provide stability with respect to the selected rate and a quick reaction to fast changing channel conditions.

<table>
<thead>
<tr>
<th>Rate (Mbps)</th>
<th>( P_{\text{ORI}} )</th>
<th>( P_{\text{MTL}} )</th>
<th>ewnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>50.00</td>
<td>N/A</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>14.34</td>
<td>39.32</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>18.61</td>
<td>28.68</td>
<td>20</td>
</tr>
<tr>
<td>18</td>
<td>13.25</td>
<td>37.22</td>
<td>20</td>
</tr>
<tr>
<td>24</td>
<td>16.81</td>
<td>26.50</td>
<td>40</td>
</tr>
<tr>
<td>36</td>
<td>11.50</td>
<td>33.63</td>
<td>40</td>
</tr>
<tr>
<td>48</td>
<td>4.70</td>
<td>23.00</td>
<td>40</td>
</tr>
<tr>
<td>54</td>
<td>N/A</td>
<td>9.40</td>
<td>40</td>
</tr>
</tbody>
</table>

Figure 4.4: RRAA implementation parameters for IEEE 802.11a.

The adaptive RTS algorithm is designed in the following way. The algorithm maintains a window and a counter \( R\text{TS}w\text{nd} \) and \( R\text{TS}c\text{ounter} \). The RTS/CTS exchange is used when the \( R\text{TS}c\text{ounter} \) is greater than 0. \( R\text{TS}w\text{nd} \) denotes the window within all data frames are transmitted using the RTS/CTS exchange. \( R\text{TS}w\text{nd} \) is initially set as 0. It is then adapted to the estimated collision level as follows. When the last frame was lost without RTS/CTS exchange, \( R\text{TS}w\text{nd} \) is incremented by one because the lost was potentially caused by a collision. When the last frame was lost with RTS, or succeeded without RTS/CTS exchange, \( R\text{TS}w\text{nd} \) is halved because the last frame clearly did not experience collisions. When the last frame succeeded with RTS/CTS exchange, \( R\text{TS}w\text{nd} \) is kept unchanged. \( R\text{TS}c\text{ounter} \) is decremented by one after every transmission with RTS/CTS exchange.

The basic RRAA and the adaptive RTS algorithm are combined as follows. After the transmission of a data frame the outcome is fed as feedback into the algorithm. First the A-RTS is run and update the \( R\text{TS}w\text{nd} \). If the RTS/CTS exchange succeeded, RRAA-BASIC is run. When the \( R\text{TS}w\text{nd} \) exceed a fixed threshold of 3, which indicates severe collisions, the hardware retry mechanism with a more robust physical data rate is disabled and the rate suggested by the RRAA-BASIC is used for the retry.

### 4.10 Receiver Based Auto Rate

The Receiver Based Auto Rate (RBAR) algorithm was published by G. Holland et al in 2001 [12]. It changes the RTS/CTS frame exchange in the IEEE 802.11 standard. In this phase the sender and receiver exchange information about the measured signal-to-noise (SNR) of the last transmission. A brief introduction into the physical model, that describes the wireless channel, is given in the background chapter.

Before each data transmission a RTS/CTS exchange should take place. The source initiates the rate adaptation by sending a RTS frame. The destination calculates
the best data rate for the upcoming transmission based on the measured SNR from the RTS frame and a set of SNR measurements from previous transmissions. The result is sent back to the source via the CTS frame. The source sends the data frame with the date rate the destination has advertised. This calculation requires a common model of the wireless channel at all stations. In the publication the authors describe the computation of thresholds that determine the fastest and least robust physical data rate for the next transmission, given the measured SNR. This computation is based on the Rayleigh fading model as described in the background chapter.

The modifications to the RTS and CTS frames make this approach completely incompatible with existing IEEE 802.11 devices. The RTS/CTS exchanges for every data frame take a lot of channel capacity. Another problem is the fact that not all wireless hardware supports the measurement of the SNR ratio, because it is not mandatory. But even if it can not be implemented due to the lack of backward compatibility, the RBAR algorithm has an important role in academics as a theoretical performance reference.

4.11 SNR-based Auto Rate for Multicast

The SNR-based Auto Rate for Multicast (SARM) was published by Y. Park et al in 2006 [30]. The algorithm is based on the RBAR link adaptation described in the previous section.

The algorithm uses the periodical beacon frame, transmitted by the access point every 100\,ms, to measure the SNR at each associated station. The stations that are member of a multicast group reply to the access point with an update of the measured SNR when necessary. To implement this the beacon is extended with the previous minimum SNR value and the station that experienced this SNR. A station transmits feedback to the access point when the measured SNR is weaker than the old minimum announced in the beacon or when the station is the announced weakest station. In the initialization stage the beacon does not contain SNR information and all members of a group transmit feedback.

The access point maintains channel state tables for each multicast group. To enable this, the authors suggest the use of IGMP snooping at the access point. IGMP snooping is introduced in the background chapter and enables the access point to learn about the multicast group membership of its associated stations. The access point chooses the physical data rate for the next transmission addressed to a group based on the corresponding channel state table.

The physical data rate is selected using thresholds as described for the RBAR algorithm. The derivation of these thresholds differ from the RBAR in the following way. The SARM is based on the Ricean fading model. This model equals the Rayleigh model, introduced in the background chapter, but assumes the presents of
a dominant propagation along a line-of-sight between sender and receiver.

In addition to this, the authors propose to turn off the error detection in the link layer. Every frame contains a check sequence that can be reproduced from the remaining frame when no error occurs. By turning this off frames received with errors are forwarded up to the network layer. The result is a higher throughput and a lower packet loss rate. This approach exploits that the group addressed multicast traffic uses UDP packets in the network layer and that the audio/visual application running in the application layer provide additional feedback to compensate UDP packet losses.

4.12 SNR-based Link Adaptation for More Reliable Groupcast

Our approach for a SNR-based link adaptation is based on the More Reliable Groupcast (MRG) Block Acknowledgment introduced in IEEE 802.11aa. The MRG Block Ack is an extension for the already existing Block Ack policy to provide feedback for group addressed transmissions.

The access point transmits a block of data frames to a group address. The stations that are group member transmit feedback about the reception of group addressed traffic after the access points request. The access point determines the size of the next block based on the feedback provided by the members and the physical data rate based on the measured SNR of the feedback transmissions.

The chapter on the implementation provides a detailed description of our approach. The chapter describes the implementation using the ns-3 network simulator and its limitations due to missing features in the given simulation.
Chapter 5

Admission Control in IEEE 802.11

Admission Control names different approaches to schedule traffic in order to maintain Quality of Service (QoS) properties. Since these transmissions have to share a limited channel, a transmission with lower requirements might need to be deferred to ensure that the tighter requirements of another is met.

These requirements are normally a throughput that must not be under-run, an end-to-end delay that must not be overshot and thresholds that limit the allowed packet loss and jitter.

5.1 Pre and Post Admission Control

Admission control is divided into two major groups. The pre and the post admission control. Pre admission control takes place before the traffic affects the channel. Post admission control takes place after the traffic has affected the channel.

An example for pre admission control in IP networks is DiffServ. Based on predefined priority rules the router decides if the packet is dropped or forwarded to a network and how long it may be deferred if other traffic is present. This implements a pre admission control on a per packet and per hop basis. The admission control takes place before the traffic affects a network. Using DiffServ the network administration can control how much traffic is allowed on a network.

The Transport Control Protocol (TCP) implements a flow control and a congestion control. The flow control is to prevent the sender from overwhelming the receiver with data. The congestion control reduces the transmission rate at the sender, if the data will not reach the receiver due to a congested network. This is an example for post admission control. The admission control takes place after the traffic has affected the network. Both controls react on packet loss due to buffer overflow, the flow control at the receiver and the congestion control on the network. The congestion control reduces the transmission rate at the sender to reduce the
chapters and congestion on the network.

5.2 User-Centric QoS Admission Control

In this section we present a proposal for a QoS admission control system. It combines pre and post admission control. The traffic is identified using tags per IP packet. The admission control is implemented at the application layer. It configures the IEEE 802.11 QoS access categories and filters traffic using the operation systems IP packet filters. The admission control decisions are based on rules defined by a central bandwidth broker and on a decentralized monitoring of the queued traffic.

A. Spenst has presented in [27] his User-Centric QoS admission control. In his approach the user can define and change for every application the priority the traffic of a specific application will have via a graphical user interface at run-time. These updates are processed by a bandwidth broker that by itself informs all stations in the network about their assigned bandwidth.

A central manager provides the different stations with profiles about the traffic admission policy. Some of these profiles are pre-defined and represent the corresponding application needs. Some are generated according to the current state of the admission control. At the stations the profiles are used to ensure that only the allowed traffic is transmitted on the network, thus to limit or to block the admission of not allowed traffic.

Figure 5.1: Diagram of the User-Centric QoS by A. Spenst et al. Image: Spenst

Figure [5.1] shows a diagram of the User-Centric QoS. The user chooses a profile (1). Profiles are predefined representations of traffic preferences for applications. Now
the QoS Manager updates its database. The QoS Manager is the central bandwidth broker in the User-Centric QoS. It maintains a database with users, applications, application types, profiles that map from application to importance, and running applications. If the user starts an application, the local process monitor detects the new application (3) and the operation system requests QoS setup from the manager and announces the new application (4). The manager updates its database (5) and determines the new traffic policy (6). As last step, the manager broadcasts the update to all stations, so that priorities on outgoing packets can be updated at all stations in the network (7).

5.3 EDCA Admission Control

Together with the EDCA channel access the traffic stream mechanism was introduced with IEEE 802.11e. The traffic stream mechanism is used to transmit parameters for admission control. In this section we give a description how an access point can realize admission control by distributing individual TXOP durations to the different associated non-AP stations. The TXOP is the duration a station may occupy the wireless channel without interruption in order to transmit traffic.

The IEEE 802.11 does not define an algorithm for computing the current admitted traffic load or to determine if an additional traffic stream is still feasible. In this section we present the algorithm suggested by the IEEE 802.11 and show how this approach can be extended for the needs of audio/visual streaming applications.

5.3.1 Traffic Stream Setup

The traffic stream mechanism is an optional part of the IEEE 802.11 standard. A non-AP station can specify its traffic stream requirements and requests the access point to provide the necessary access to the wireless channel by sending an ADDTS Request management action frame. This traffic stream may be up-link from the station to the access point or down-link from the access point to the station or bidirectional. The ADDTS frame contains a traffic specification element (TSPEC). In the TSPEC the traffic stream requirements are expressed in terms of data rate, delay bounds, packet size, and other parameters. After the end of the transmission the non-AP station informs the access point by sending a DELTS frame. The standard does not define how the TSPEC is generated. The implementation may use information provided by a user interface at the station or information obtained by protocols from other layers. The traffic stream mechanism acts as a bridge connecting the Network Layer QoS mechanisms like DiffServ with the Link Layer QoS mechanisms.

The access point answers with an ADDTS Response frame that specifies the medium time. If the access point accepts the traffic stream, the TSPEC in the ADDTS Response specifies the medium time. The medium time is the fraction of the TXOP
that may be used for the traffic stream. The access point calculates the existing load based on the current set of issued TSPECs. Based on the current conditions, the access point may accept or deny the new TSPEC request. If the access point accepts the additional traffic stream, the ADDTS Response contains a Schedule Element. The ADDTS frames can also contain a traffic classification element (TCLAS). This optional element contains the set of parameters necessary to identify incoming MSDUs with a particular traffic stream.

The IEEE 802.11aa defines extensions for the use of the traffic stream mechanism for group addressed traffic. It defines a Stream Classification Service (SCS) that is going to provide the possibility of expressing priorities among traffic streams within an access category. In addition to the ADDTS frames the MRG Request and MRG Response frames defined in the IEEE 802.11aa also enable the optional transmission of TSPEC and Schedule Element.

During the association the access point provides the initial parameters for the configuration of the EDCA access categories. These parameters are given in the so-called EDCA Parameter element. This element also contains the maximal allowed TXOP per access category. The access point can provide updated TXOP durations to the individual stations. The EDCA Parameter element is distributed via the periodical beacon frame or via the individual association or re-association frames. The remaining parametrization of the access categories remain constant for all stations.

In addition to the EDCA parameter exchange every station can announce traffic to the access point. The flag Admission Control Mandatory in the EDCA Parameter element signals if the traffic stream setup is mandatory before using an access category. If admission control is mandatory for an access category, it is mandatory for all access categories with higher priority. For example if it is mandatory for access category video, it is also mandatory for access category voice. The use of admission control is not recommended for the access categories with priority best effort or background. Since the traffic stream mechanism is optional, the minimal support an access point must provide is to advertise that no admission control is needed for using the high priority access categories.
5.3. EDCA ADMISSION CONTROL

5.3.2 Parameter Derivation

The derivation of the parameters describing the traffic stream properties and needs are not part of the IEEE 802.11 standard. Together with the algorithm that realizes the admission control the derivation of these parameters is up to the actual implementation of the access point. The following presents the solution suggested in the standard and points out possible extensions to cover the needs of audio/visual streaming applications.

Medium Time

The admission control using the traffic stream is defined for HCCA and EDCA. We only present the one mode available in EDCA. This mode is for traffic with a constant bit rate. The parameters used in the TSPEC element are the nominal MSDU size, mean data rate, the minimal physical data rate and the surplus bandwidth allowance denoted as $sba$. The medium time is set by the access point in the TSPEC contained in the ADDTS Response. The medium time can be computed using the formula

$$ medium \ time = sba \cdot pps \cdot T_{\text{hold}} \$$

where the term $pps$ denotes the packets per second and is defined as

$$ pps = \frac{1}{\text{MSDU size}} \cdot \frac{\text{mean data rate}}{8}. $$

The value $T_{\text{hold}}$ is defined as

$$ T_{\text{hold}} = TXTIME(\text{MSDU size, physical data rate}) + SIFS + ACKduration $$

where $TXTIME$ is the time it takes to transmit a MPDU given the minimal physical data rate and the nominal MSDU size. The formula does not take explicitly into account how the fragmentation of a MSDU into several MPDUs or the aggregation of multiple MSDUs into an A-MSDUs affects the transmission time. The formula implicates that the MSDU size is constant. This holds true for most applications with a constant bit rate. The formula also does not express explicitly the effect of the different acknowledgment policies on the transmission time.

The surplus bandwidth allowance parameter expresses the amount of time needed to transmit packets in excess to compensate from packet losses. Regardless of the possibility to adapt the transmission to the changing conditions of the wireless
channel using the different physical data rates is the wireless channel an erasure channel with a probability $p_e$ that a transmission is lost. This probability $p_e$ is also known as packet error rate (PER). We assume in this example that the channel causes errors independently from packet to packet and the error probability is the same for all packets of the same size. Such a channel is called an independent, identically distributed error channel (iid). The capacity $C$ of an erasure channel is given by $C = 1 - p_e$. The packets that must be transmitted in excess to compensate from this loss is the redundant information needed to provide a reliable transmission.

$$RI_{\text{reliable}} = \frac{p_e}{1 - p_e}$$

For a packet error rate of $p_e = 10\%$ the surplus bandwidth allowance must be 1.1111 to provide enough redundancy to compensate lost transmissions to give an example. An increased number of transmissions to compensate from losses means that a bigger part of the given channel capacity is used by the traffic stream and also that the time, needed for this additional transmissions, increases the introduced delay. This is critical for audio/visual streaming applications. This kind of applications has strict delay bounds but can tolerate some packets to be lost. To take advantage of the applications properties, an erasure channel model with a fault tolerance can be used. An erasure channel with some residual error rate can be modeled by a cascade of two erasure channels. The erasure channel with fault tolerance can be derived as in [11]. The overall loss rate is given by the equation

$$p_e = 1 - (1 - p[1])(1 - p[2]) = 1 - (1 - p[1])(1 - p_{\text{res}})$$

where $p[1]$ is the loss rate of the first erasure channel and $p[2]$ the loss rate of the second. The second channels probability can be interpreted as the residual erasure rate $p_{\text{res}}$. This equation can be solved for $p[1]$

$$p[1] = \begin{cases} \frac{p_e - p_{\text{res}}}{1 - p_{\text{res}}} & p_e \geq p_{\text{res}} \\ 0 & \text{else} \end{cases}.$$ 

By this construction the first channel can be interpreted as being an erasure channel with a residual output loss rate of $p_{\text{res}}$. The redundant information needed to achieve an overall PER $p_{\text{res}}$ the traffic stream application can tolerate is now determined by the formula

$$RI_{\text{res}} = \frac{p[1]}{1 - p[1]} = \frac{p_e - p_{\text{res}}}{1 - p_e}.$$ 

The capacity of an erasure channel with some residual error rate $p_{\text{res}}$ is given by
5.3. EDCA ADMISSION CONTROL

\[ C = 1 - p[1] = \frac{1 - p_e}{1 - p_{res}}. \]

The reduction of reliability is expressed as a reduction of the denominator in the computation of the channel capacity. Because of this a higher allowed residual error rate provides a higher available channel capacity. In addition to this the reduction of possible retransmissions decreases the possible introduced delay.

The transmissions needed in excess to provide the residual information can be transmitted in two ways. The first is forward error correction (FEC). Forward error correction means that the additional packets are transmitted before any feedback. In case of an error the recipient tries to compensate from the loss using the additional packets. The second is to transmit additional packets only in case of feedback that requests this transmissions to compensate from errors. This is called automatic repeat query (ARQ). FEC and ARQ introduce both additional delay and feedback. FEC introduces only a little additional delay with the generation of the redundant packets. But it introduces a lot of overhead since these additional packets are always transmitted. ARQ introduces less overhead than FEC, since the additional packets are only transmitted in case of an error. But ARQ introduces a greater delay than FEC, since the additional packets are first transmitted when the feedback reaches the originator. The construction of an adaptive hybrid error correction (AHEC) that determines an optimal combination of FEC and ARQ that is feasible within the available channel capacity and delay bounds is described by M. Gorius et al in [28].

The computation of the medium time must also take into account the characteristics of the IEEE 802.11 standard. The fragmentation of a MSDU into several MPDUs is relative easy to express, since the usual implementation defines a static MSDU size as threshold for the fragmentation. Every MSDU with a size above this threshold is fragmented into several MPDUs. The effects of the A-MSDU as defined by the Project IEEE 802.11n Task Group or the Block Ack and MRG Block Ack as defined by the Project IEEE 802.11aa Task Group are more challenging. The number of MPDUs transmitted in one block is not fixed. This block size is limited by the available buffers in the transmitting and receiving stations. In practice the block size is smaller than this limit, due to the link adaptation that tries to adapt the physical data rate and the block size to the changing channel conditions. An example for this behavior is the SNR-based link adaptation we present in this work. Thus a robust and simple approach to this problem is to assume the worst case, the transmission using the minimal allowed physical data rate and assuming that only one MSDU is transmitted in one block or A-MSDU. This is what the proposed approach implicitly does.
Schedule Generation

The traffic stream mode defined for use with EDCA uses the nominal MSDU size, mean data rate, the minimal physical data rate and the surplus bandwidth allowance parameter fields in the TSPEC element. To generate the schedule the medium time and service interval is computed by the access point. Since the EDCA mode is for use with constant bit rate streams the MSDU size and the data rate can be assumed to be constant.

Figure 5.3: Example of a schedule for one stream. The starts of each TXOP are at most the duration of one service interval apart, to provide the mean data rate. Image: IEEE

The service interval $T_s$ can be given among other parameters in the TSPEC element for some HCCA modes. The service interval is the time between two transmissions belonging to the same traffic stream. In the EDCA mode the service interval is computed by the access point. Usually it is defined as fraction of the packet size and the data rate.

Figure 5.4: An example schedule with 3 streams of equal size. Every stream is scheduled with the same service interval and the same medium time. Image: IEEE

For the TXOP duration the scheduler first computes the number of data frames $N_i$ that arrive during one service interval for stream $i$ with data rate $\rho_i$ and MSDU size $L_i$

$$N_i = \left\lceil \frac{T_s \cdot \rho_i}{L_i} \right\rceil.$$  

Then the TXOP duration is calculated as maximum of the time needed to transmit $N_i$ data frames at the physical data rate $R_i$

$$TXOP_i = \max \left( \frac{N_i \cdot L_i}{R_i} \right).$$

Then the TXOP duration is announced to the non-AP stations in the Schedule element contained in the ADDTS Response frame. The figures 5.3, 5.4 and 5.5 show a schedule for up to three different traffic streams with equal properties. In the
5.3. EDCA ADMISSION CONTROL

Figure 5.5: In this example the stream \( j \) is dropped and the remaining streams are reallocated. Image: IEEE

example all streams need the same service interval and the same TXOP duration. The traffic streams in this example can be interpreted as IPTV streams. If an additional stream is transmitted, the service interval and TXOP duration need by every streams remains constant. But the usage of the channel increased, with every additional stream.

5.3.3 Admission Control Unit

The decision if an additional traffic stream is admitted is done by a so-called admission control unit. This unit computes the channel usage based on the already admitted traffic streams. The additional stream is feasible, if the inequality

\[
\frac{TXOP_{k+1}}{T_{s_{k+1}}} + \sum_{i=1}^{k} \frac{TXOP_{i}}{T_{s_{i}}} \leq \mu
\]

is satisfied. The IEEE 802.11 standard propose to define \( \mu \) as fraction with the time used for traffic using EDCA as nominator and the beacon interval as denominator. This makes sense for the admission control of traffic using HCCA. A more general bound for \( \mu \) was found by Liu and Layland in [17]. For the so-called rate monotonic scheduling of multiple programs in hard real-time environments using a single processor they found that

\[
\mu = k(\sqrt{2} - 1).
\]

When \( k \) tends towards infinity the expression will tend towards

\[
\lim_{n \to \infty} n(\sqrt{2} - 1) = \ln 2 \approx 0.693147 \ldots
\]

Applied on the admission control unit this means that approximately 70% of the time can be used for high priority traffic streams and 30% for the traffic with lower priority. The result from Liu and Layland can be applied, since both problems describe the scheduling of periodical tasks with a constant interval between each execution using a shared resource among all tasks. The inequality defines the upper bound of tasks that are feasible together without violating the constant scheduling intervals.
Chapter 6

The ns-3 Network Simulator

In this chapter we provide an introduction to the ns-3 network simulator. We use the ns-3 network simulator as basis for our implementation of MRG Block Ack. Based on this implementation we implement a SNR based link adaptation for the MRG Block Ack mechanism to realize an efficient and reliable transmission of group addressed traffic. In a next step, we run experiments with this simulation, to test it and to determine an optimal parametrization.

6.1 Overview

The ns-3 [23] is the designated successor of the popular ns-2 network simulator [22]. It is a discrete-event simulator for Internet systems, targeted primarily for research and educational use. The work on ns-3 started in 2006. A first stable version was released in June 2008. Our implementation is based on version 3.6 from October 2009. The latest stable release is version 3.7.1 from March 2010. The ns-2 simulator is written in the two programming languages OTcI (MIT Object Tcl) and C++. For ns-3 the scripting language OTcI is replaced with the more widespread scripting language Phyton. The design goal is to provide a network simulator that enables the user to set up a simulated network in a fast and easy way by writing a script file. This script hides most of the setup details from the user. The scripted part of the simulation will call the part written in C++. This part is the actual simulation and written in C++ since the compiled machine code provides a much faster execution. Figure 6.1 provides an overview of the components written in C++. On top are the helper wrapper. These helpers are written in C++ and provide the interface to the Phyton scripting language. Of course one can also write a C++ program calling these helpers. Using this approach the user needs no programming language except C++. The ns-3 is used although it does not provide all functionality needed. In an evaluation, testing different network simulation software for educational and scientific use, the ns-3 network simulator showed to be the best fitting solution.

The main components of ns-3 are shown in Figure 6.1. A simulation in ns-3 con-
Chapter 6. The NS-3 Network Simulator

Figure 6.1: The global architecture of ns-3. Image [23].

The NS-3 network simulator consists of nodes, devices, protocol stacks, applications and at least one channel. These nodes, devices and applications are organized in containers. Channels are implemented as special containers. A channel is a container that provides some form of message passing among its members. The different components can be instantiated and parametrized by helper wrappers. With the use of containers this can be done for a group of common instances at once. The usual way is to create a NodeContainer with the nodes that own a device connected to a common channel. In the next step one creates a helper for the channel and generates the devices that connect the nodes to the channel by applying the helper on the NodeContainer. The devices are accessible via a specific device container, that also serves as the channel for the simpler channel models, like the PointToPoint model. The protocol stack and the applications are installed in a likewise way. Figure 6.2 shows two nodes with complete network stack, connected over a channel. A Node can be in multiple NodeContainers and can have multiple devices, in order to connect it to multiple channels. This is needed to simulate router and bridges in ns-3. ns-3 also provides for the applications an POSIX-like API to make it easier to connect real applications to the simulation.

Some of the devices provided by ns-3 (version 3.7) are:

**PointToPoint** This is a very simple point to point data link connecting exactly two nodes with each other. This can be viewed as equivalent to a full duplex RS-232 or RS-422 link with null modem and no handshaking.

**CSMA Device** This model provides a simple wired bus model. It does not model a real physical network, like IEEE 802.3, but it provides some sort of collision
6.2 IEEE 802.11 IMPLEMENTATION

Figure 6.2: The node architecture in ns-3. Image [23].

detection for multiple devices on one channel.

**Bridge Device** This implements the behavior of an IEEE 802.1D bridge device. It can be used for instance to connect multiple networks using the CSMA Device model in order to separate the collision domains.

**Tap Device** This device provides connection to a Linux “Tap” device. The Linux “Tap” device is a virtual network adapter. It provides routing of packets from and to real network adapters. This enables ns-3 to connect to real networks and applications.

**Wifi** The IEEE 802.11 implementation in ns-3. It is presented in the next section. It provides a device that simulates the behavior of a wireless network adapter. Different 802.11 MAC and physical Layer implementations can be used to implement different wireless communication behavior.

For the implementation and simulation of the IEEE 802.11aa and the SNR-based link adaptation we extend the Wifi device.

6.2 IEEE 802.11 Implementation

This section provides a very condensed overview of the implementation of the IEEE 802.11 wireless LAN in ns-3. We focus on the implementation of the infrastructure mode with support for QoS transmissions. The description mixes the naming of abstract base classes and derived classes where it helps to provide an overview of the interaction between the components. Most of the details are omitted, except those which are subject of later changes by us.
6.2.1 The WifiNetDevice

The WifiNetDevice provides a skeleton for the different components a wireless network interface consists of. It provides an interface to the node in ns-3. The current implementation is roughly divided in 4 levels of models by the ns-3 documentation.

**The physical layer models** They provide the simulation of the wireless transmission reception and propagation. To provide this functionality they are closely related with the simulation of the wireless channel and the mobility model for the nodes.

**MAC low models** They implement the DCF and EDCAF functionality. At this level the flow control with the acknowledgment of transmitted data frames and the link adaptation takes place.

**MAC high models** They implement the MAC-level management functionality. For instance the beacon generation, probing, and association state machines. It exists in two different variants. One for the Ad Hoc mode and one for the infrastructure mode.

**Rate control algorithms** The MAC low models use an instance of an rate control algorithm to implement the link adaptation.

Figure 6.3 provides an overview about the collaboration and information flow in the IEEE 802.11 model.

**WifiNetDevice** This is the central class of the Wifi Device in ns-3. It does not implement functionality, but provides a common interface to ns-3 and for the various components of the 802.11 implementation. Is is normally instantiated by a helper wrapper.

**WifiMac** The WifiMac is an abstract base class. During the construction of the WifiNetDevice the helper wrapper is given an instance of a derived class. The type and functionality of the WifiNetDevice is defined by this instance. Currently exists 6 derived classes, that implement Ad Hoc and infrastructure mode, with and without support for QoS.

- **AdhocWifiMac** does not perform any kind of beacon generation, probing, or association. This class represents the minimal subset of functions that must be provided to ensure the function of the WifiNetDevice.

- **NqstaWifiMac** provides automatic association with an access point in range, and re-association if to many beacons are lost.

- **NqapWifiMac** implements an active beacon generation and keeps state about associated non-AP station in collaboration with the WifiRemoteStation-Manager.

- **QadhocWifiMac** is the QoS version of the AdhocWifiMac. It provides support for QoS traffic in 4 different access categories.
Figure 6.3: The WifiNetDevice implements the IEEE 802.11 in ns-3. This diagram shows the information flow between the different components.

**QstaWifiMac** is the QoS version of the NqstaWifiMac. It provides support for QoS traffic in 4 different access categories.

**QapWifiMac** is the QoS version of the NqapWifiMac. It provides support for QoS traffic in 4 different access categories.

**DcaTxop** This class is derived from the Dcf base class. It is instantiated by the non-QoS high level WifiMac classes. Together with the MacLow and the DcfManager it implements the CSMA/CA.

**EdcaTxopN** The EdcaTxopN is also derived from the Def base class as the DcaTxop. The high level Mac implementation instantiate four instances of this class, one for each access category. It implements functions like fragmentation and aggregation of MPDUs. Together with the DcfManager and the MacLow class it implements the CSMA/CA in QoS Wifi. After a successful transmission it resets its contention window in the DcfManager or increases it after a failure. It holds a packet forwarded to the lower layer and blocks until the reception of an acknowledgment or a timeout is signalized by MacLow. Packets arriving meanwhile are queued. If necessary, it initiates the retransmission of a frame. It requests access to the channel via the DcfManager.

**DcfManager** Here the CSMA/CA is implemented in the wifi simulation. It controls the four instances of EdcaTxopN in a WifiNetDevice with QoS.
QoS it controls one instance of the DcaTxop class. It also controls in addition and independent from this instances an extra instance of the DcaTxop for the beacon generation in the QapWifiMac and the NqapWifiMac. The instances of EdcaTxopN and DcaTxop are created at the high level Mac implementations together with the DcfManager. After the creation the high level Mac registers this instances at the DcfManager. The manager uses the WifiPhy instance for physical carrier sensing and the MacLow for virtual carrier sensing. If an registered instance of the DcaTxop or EdcaTxopN requested access to the channel, it uses the carrier sensing to run the CSMA/CA algorithm. If it gains access to the channel it calls the according registered instance.

**MacTxMiddle** In this class the sequence number counters for the different destinations are maintained.

**WifiRemoteStationManager** The manager for the remote stations is an abstract base class. Instances of the derived classes are given to the helper wrapper. This wrapper then sets the callbacks to the instance of the WifiMac class. The WifiRemoteStationManager maintains instances of the WifiRemoteStation class. For every new destination an instance is created. All broadcast and multicast traffic is managed by one separate instance of the WifiRemoteStation class. An instance of WifiRemoteStation lives until the ns-3 simulation terminates.

**WifiRemoteStation** This is the abstract base class that implement the framework for link adaptation. In addition to this, it keeps track of the association state, if used with an NqapWifiMac or QapWifiMac. It provides an interface for the MacLow class, to be informed on successful or lost transmissions and to provide the data rate for the next transmission. The strategy used to determine the data rate for the next transmission is selected by the different classes derived from WifiRemoteStationManager. In ns-3 (version 3.7) the following algorithms are given:

- **ArfWifiManager** implements the auto rate fallback algorithm.
- **AarfWifiManager** implements the adaptive auto rate fallback algorithm.
- **IdealWifiManager** implements a mechanism based on the measured SNR of the last transmission, similar to the RBAR algorithm.
- **OnoeWifiManager** is an implementation of rate control algorithm developed by Atsushi Onoe. This algorithm has been used by the madwifi driver. No further publication is known.
- **AmrrWifiManager** implements the adaptive multi rate retry algorithm presented by Lacage et al in [16] together with the AARF.
- **CaraWifiManager** implements the Collision-Aware Rate Adaption algorithm from: J. Kim et al [14]. It uses RTS/CTS as probes after consecutive transmission errors.

---

1. [http://www.nsnam.org/doxygen-release/classns3_1_1_onoe_wifi_manager.html](http://www.nsnam.org/doxygen-release/classns3_1_1_onoe_wifi_manager.html)
AarfdWifiManager implements an extension to the AARF algorithm that makes use of the RTS/CTS mechanism to decide between transmission errors due to collisions or due to changed channel conditions. Maguolo, Lacage, Turletti [19].

In the link adaptation chapter we introduce this algorithm in greater detail.

MacLow Depending on the transmission parameters MacLow initiates RTS/CTS transmission or waits for the reception of an ACK frame. It gets the transmission mode for the next transmission from the WifiRemoteStation corresponding to the transmissions destination. It initiates every transmission by forwarding the packet to the WifiPhy. If an acknowledgment is expected in a defined delay, like an ACK frame after a data frame or an CTS frame after an RTS frame, it maintains the reception or timeout of this expected reply with a set of timers. This timers are set on transmission and canceled on the reception of the expected reply. If a timeout occurs, the appropriate method is called to handle the event. The MacLow signals the success or failure of the transmission to the WifiRemoteStation of the current destination and the EdcaTxopN. The WifiRemoteStation to provide the link adaption. The EdcaTxopN is called to signal the complete transmission or failure of the current transmission. The EdcaTxopN then releases its lock, and starts the transmission of the packet or the retransmission of the current. The MacLow forwards successfully received packets to MacRxMiddle.

MacRxMiddle This class handles detection of duplications, and reconstruction of fragmented MSDUs. It gets received packets from MacLow and forwards them to the used instance of WifiMac.

WifiPhy models the wireless radio. The WifiPhy models by default a 802.11a channel, in terms of frequency, data rate, and bit rates, and interacts with the propagation loss model and the propagation delay model found in the channel. The implementation is done in the YansWifiPhy class. Every physical data rate defined by a standard like the 802.11a is represented in an instance of the WifiMode class. The name YansWifiPhy is a reference to the model described in [15] from which this implementation is taken. The YansWifiPhy can be in one of the following states.

IDLE The physical layer is IDLE.

CCA_BUSY The physical layer has sense the medium busy through the CCA mechanism.

TX The physical layer is sending a packet.

RX The physical layer is receiving a packet.

SWITCHING The physical layer is switching to another channel.

When the first bit of a new packet is received while the WifiPhy is not in state IDLE, the received packet is dropped. Otherwise, if the received energy of the first bit of this new signal is computed and compared against the energy detection threshold. If the energy of the packet is higher, then the WifiPhy moves
 CHAPTER 6. THE NS-3 NETWORK SIMULATOR

to RX state and schedules an event when the last bit of the packet is expected
to be received. Otherwise, the WifiPhy stays in IDLE state and drops the
packet. In this model CCA becomes busy when the aggregation of all signals
as tracked by an internal interference helper is higher than a preset threshold
and the WifiPhy can not synchronize on a packet. The model becomes idle
if the aggregation of all signals drops below a threshold. A transmission can
start only if the model is in the idle state before and no channel switching takes
place.

In the interval between the reception of the first and the last bit of the packet,
the transmission power is calculated with a path-loss propagation model. Out-
side the reception interval the energy of the received signal is assumed to be
zero. From the received energy the signal-to-noise pulse interference (SNIR)
is computed. From the SNIR function the bit error rates (BER) is derived for
BPSK and QAM. From this is the SNR and the probability that the packet is
received correctly is computed. This computation is done via an common error
rate model named YansErrorRateModel and individual power loss models that
describe the individual channel conditions.

The setup of the different power loss models is done via the helper wrapper. The
parameters necessary for the computation, most important the distance,
are computed in the mobility model and passed via the WifiChannel into the
YansWifiPhy. The following propagation loss models are available in the Yan-
sWifiPhy. They can be combined to model the channels physical characteristics. Each model computes the loss in dBm and the different degradations
are added up. The following propagation loss models are available. The description
is taken from the source code\[1\]:

RandomPropagationLossModel In this model the propagation loss follows
a random distribution.

FriisPropagationLossModel The Friis propagation loss model was first de-
scribed in "A Note on a Simple Transmission Formula", by "Harald T.
Friis". The original equation was described as:

\[ \frac{P_r}{P_i} = \frac{A_r A_t}{d^2 \lambda^2} \]

with the following equation for the case of an isotropic antenna with no
heat loss:

\[ A_{isotr.} = \frac{\lambda^2}{4\pi} \]

The final equation becomes:

\[ \frac{P_r}{P_i} = \frac{\lambda^2}{(4\pi d)^2} \]

Modern extensions to this original equation are:

\[ P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \]

\[ ^{2}\text{File src/devices/wifi/propagation-loss-model.h} \]
With:
\( P_r \) reception power (W)
\( P_t \) transmission power (W)
\( G_t \) transmission gain (unit-less)
\( G_r \) reception gain (unit-less)
\( \lambda \) wavelength (m)
\( d \) distance (m)
\( L \) system loss (unit-less)
This model is invalid for small distance values. The current implementation returns the transmission power as the reception power for any distance smaller than a minimal distance.

**LogDistancePropagationLossModel** This model calculates the reception power with a so-called log-distance propagation model:

\[
L = L_0 + 10 n \log_{10} \left( \frac{d}{d_0} \right)
\]

where:
\( n \) the path loss distance exponent
\( d_0 \) reference distance (m)
\( L_0 \) path loss at reference distance (dB)
\( d \) distance (m)
\( L \) path loss (dB)

When the path loss is requested at a distance smaller than the reference distance, the tx power is returned.

**ThreeLogDistancePropagationLossModel** A log distance path loss propagation model with three distance fields. This model is the same as LogDistancePropagationLossModel except that it has three distance fields: near, middle and far with different exponents.

Within each field the reception power is calculated using the log-distance propagation equation:

\[
L = L_0 + 10 \cdot n_0 \log_{10} \left( \frac{d}{d_0} \right)
\]

Each field begins where the previous ends and all together form a continuous function.

There are three valid distance fields: near, middle, far. Actually four: the first from 0 to the reference distance is invalid and returns the transmission power in dBm.

\[
\begin{array}{cccccccc}
0 & \cdots & d_0 & \cdots & d_1 & \cdots & d_2 & \cdots & \infty \\
=0 & n_0 & n_1 & n_2 & \end{array}
\]

Complete formula for the path loss in dB:
\[ L = \begin{cases} 
0 & d < d_0 \\
L_0 + 10 \cdot n_0 \log_{10}(\frac{d}{d_0}) & d_0 \leq d < d_1 \\
L_0 + 10 \cdot n_0 \log_{10}(\frac{d}{d_0}) + 10 \cdot n_1 \log_{10}(\frac{d}{d_1}) & d_1 \leq d < d_2 \\
L_0 + 10 \cdot n_0 \log_{10}(\frac{d}{d_0}) + 10 \cdot n_1 \log_{10}(\frac{d}{d_1}) + 10 \cdot n_2 \log_{10}(\frac{d}{d_2}) & d_2 \leq d
\end{cases} \]

where:

- \( L \) resulting path loss (dB)
- \( d \) distance (m)
- \( d_0, d_1, d_2 \) three distance fields (m)
- \( n_0, n_1, n_2 \) path loss distance exponent for each field (unitless)
- \( L_0 \) path loss at reference distance (dB)

When the path loss is requested at a distance smaller than the reference distance \( d_0 \), the transmission power (with no path loss) is returned. The reference distance defaults to 1m and reference loss defaults to FriisPropagationLossModel with 5.15 GHz and is thus \( L_0 = 46.67 \text{ dB} \).

**NakagamiPropagationLossModel** The Nakagami-m distribution is applied to the power level. The probability density function is defined as

\[
p(x; m, \omega) = \frac{2m^m}{\Gamma(m)\omega^m} x^{2m-1} e^{-\omega x^2} = 2x \cdot p_{\text{Gamma}}(x^2, m, \frac{m}{\omega})
\]

with \( m \) the fading depth parameter and \( \omega \) the average received power. It is implemented by either a Gamma Variable or a Erlang Variable random variable.

Like in the ThreeLogDistancePropagationLossModel, the \( m \) parameter is varied over three distance fields:

\[
0 \cdots d_1 \cdots d_2 \cdots \infty
\]

For \( m = 1 \) the Nakagami-m distribution equals the Rayleigh distribution. Thus this model also implements Rayleigh distribution based fast fading.

**WifiChannel** In this class the wireless channel is simulated. The YansWifiChannel class provides propagation delay and power loss models and is based on the same model as the YansWifiPhy class. In the YansWifiChannel the connection to the Mobility Model of ns-3 is made. The helper wrapper for the mobility model takes a node and connects it to the mobility model. The mobility model runs independent from the Wifi model. A node may be stationary, like an access point, or mobile. For the mobile nodes different movement models exist. Like random walk, way-point or different acceleration movement models. The propagation delay model is set via the helper wrapper.

The mobility models of the sending and the receiving node are passed as parameters to the delay and power loss models. The computation of the delay takes place in the channel. The following two propagation delay models are available:
ConstantSpeedPropagationDelayModel The constant speed propagation model is the default. It returns the delay based on the distance and the propagation speed. Its default is to return the delay based on the speed of light.

RandomPropagationDelayModel The random speed propagation model can be used to introduce random delay if needed.

Figure 6.4 shows the class diagram of the WifiNetDevice. The WifiPhy and WifiRemoteStationManager are parametrized and instantiated with their own helper wrappers. They are passed as parameters into the helper for the WifiNetDevice, along with the type of the WifiMac. The helper creates the WifiNetDevice and the WifiMac. Then it sets the function callbacks for the transmission of packets from and to the node and physical layer, represented by WifiPhy. The WifiMac creates on construction its subclasses and sets the function callbacks between them, to enable the transmission of packets between them.

Figure 6.4: The class diagram of the WifiNetDevice.

Message passing between the classes is realized in different ways. The first is that one class has a pointer to the other class and calls the accessor methods of the other class. This is done mainly to setup the second communication method. The function callback. A class has a member variable that takes a function pointer to the method of another class. Once mounted, the class can call the other classes method. The advantage of this approach is that it decouples the caller from the called. The called must only conform a defined interface. This enables the combination of different devices and channel types in ns-3. Thus the forwarding of data packets is implemented in this way. The third is the use of Listeners classes for message passing. The lock mechanism between the EdcaTxopN and the MacLow classes is an example for this. On transmission the EdcaTxopN send the pointer to its TransmissionListener class to the MacLow class. The MacLow class calls the appropriate method of TransmissionListener to signal for instance the successful reception of an acknowledgment for the data packet that corresponds to the transmission initiated.
by EdcaTxopN.

### 6.2.2 Initialization

In this section we describe how some of the classes are initialized. At first the helper wrappers are used to create the channel and the different WifiNetDevices. Then at run-time data is copied to new created instances or different classes.

#### Supported physical Data Rates

The WifiMac and the derived classes like the QapWifiMac provides parameters, that can be set using the corresponding helper wrapper. Using this one can parametrize the WifiNetDevice. For instance the fragmentation or the RTS/CTS threshold can be set, or the duration of an AIFS. The WifiMac then sets this parameters in the corresponding instances of DcaTxop and EdcaTxopN. The WifiPhy, or more specific the YansWifiPhy, can be parametrized to support different sets of transmission modes, like the IEEE 802.11b or the IEEE 802.11a. The YansWifiPhy creates the corresponding instances of the WifiMode for every supported data rate. The helper wrapper for the WifiNetDevice takes an instance of WifiPhy, WifiMac and a node as parameters. It links the components together, using callbacks, attaches the WifiNetDevice to the node and links it to the nodes higher layers.

The WifiRemoteStation is an abstract base class. The derived classes implement different strategies to choose the most appropriate data rate for the next transmission. The different data rates are represented in the WifiMode class. A WifiMode contains all the physical parameters like the channel bandwidth and coding parameters needed for the simulation of a transmission over a wireless channel. It is defined in the WifiPhy. After the WifiRemoteStation is created by the WifiRemoteStationManager, it is the responsibility of the WifiMac to copy all WifiModes supported by the remote station in the WifiRemoteStation. If the WifiMac is an AdhocWifiMac or QadhocWifiMac, ns-3 assumes that all stations support the same set of WifiModes. In the WifiMac that simulate infrastructure mode the supported WifiModes are part of the probe and association request and response packets and the periodical beacon.

#### Beacon Generation and Association

The access point generates a periodical beacon frame. The non-AP station keeps itself associated to the access point. This is implemented using a state machine and a watchdog mechanism in the NqstaWifiMac and QstaWifiMac. In the NqapWifiMac and QapWifiMac a dedicated instance of the DcaTxop class is instantiated for the beacon generation. It is parametrized to perform no back off, and to defer the current transmission in case of an internal collision at the access point. The actual
beacon is produced by a method that, after sending the beacon, schedules itself after on beacon interval. The dedicated DcaTxop notifies the DcfManager that it wants access to the channel. As soon as the channel is idle, the DcfManager calls the DcaTxop. Now DcaTxop forwards the beacon to MacLow and blocks until MacLow signals that the transmission is done.

At the receiving station the beacon frame is forwarded up from the MacLow to the NqstaWifiMac or QstaWifiMac that implements the non-AP station behavior. The NqstaWifiMac and QstaWifiMac try to stay associated. If their watchdog mechanism recognizes that the number of missed beacons violates a given threshold, it is most likely they are not associated anymore. In this state the non-AP station tries to get associated again. This is done by waiting for a beacon or by sending a broadcast probe request to get a probe response from an access point. If the presence of an access point is detected, the non-AP station starts association with an association request. If access point replies with a positive association response, the NqstaWifiMac and QstaWifiMac are in state associated again.

### 6.2.3 Basic Operations

In this section the basic operations of the WifiNetDevice and how the different components collaborate is described. This is not an detailed manual, but more a guide to understand the design of the WifiNetDevice and how the different concepts defined in the standard are implemented. The ns-3 does not stick to the IEEE naming of frames, MSDUs and MPDUs, but speaks most of the time about packets. We use the term packet in this section, in order to stay closer to ns-3.

**Packet Transmission and CSMA/CA**

The WifiNetDevice gets a packet from the upper layer at the node and forwards it to the WifiMac. The non-QoS WifiMac classes maintain one instance of a DcaTxop class for all traffic. The QoS WifiMac classes maintain four instances of EdcaTxopN and forward the packet according to the TID set in the header. A DcaTxop or EdcaTxopN with packets queued for transmission, notify the DcfManager that they want access to the channel. The WifiPhy signals the channels busy or idle state to the DcfManager. The back off mechanism then takes place in the DcfManager. As soon as the channel is idle, the DcfManager calls the winning DcaTxop or EdcaTxopN.

The instance that won forwards the packets to MacLow and blocks until MacLow signals that the transmission is done. The instance also fetches the next sequence number for the packet from MacTxMiddle. Together with the packet a set of transmission parameters and a TransmissionListener is forwarded to MacLow. The transmission parameters for instance determine the use of a RTS/CTS exchange before the actual transmission or if the station shall wait for an acknowledgment after
the transmission of the packet. The decision if an acknowledgment is needed, is hard-wired into the code for all packets generated in DcaTxop or EdcaTxopN. It is needed for unicast data and management packets and not for group or broadcast addressed packets. Decisions like the need of a RTS/CTS exchange or if the packets needs to be fragmented into multiple smaller before transmission is made in the WifiRemoteStation the packet is addressed to. The fragmentation is done in DcaTxop, EdcaTxopN. The EdcaTxopN implements in addition the aggregation of multiple smaller packets into a larger one.

The MacLow is called in the moment the transmission starts. At first MacLow cancels all running Events. An event is for example the timer that schedules the method call, if an acknowledgment is successfully received after the transmission of a data frame. This canceling also signals the end of a might ongoing transmission to another blocked DcaTxop or EdcaTxopN at the WifiNetDevice. Now MacLow fetches the data rate that should be used to transmit the current packet from the WifiRemoteStation. With this information the MacLow determines the transmission time for the packet. If the reception of an acknowledgment or CTS is needed, the MacLow schedules a call to its appropriate method to handle the event of the miss of that transmission. If the expected transmission is received in time, the scheduled call is canceled. So the MacLow determines the complete channel holding time needed for the transmission and stores it in the packets header. The packet is now forwarded to the WifiPhy and transmitted over the WifiChannel. If other stations WifiNetDevices on the channel receive the packet, it is forwarded up from the WifiPhy to the MacLow.

Simulation of the Wireless Channel

The wireless radio is modeled by the YansWifiPhy class. The YansWifiPhy at a WifiNetDevice takes a packet and a WifiMode from MacLow for transmission. The WifiMode contains the physical data rate at which the packet shall be transmitted. On the transmission of a packet YansWifiPhy computes the transmission duration and sets its internal state to TX. This is state is evaluated by the other components at the WifiNetDevice, like the DcfManager, to determine the channel state. Then YansWifiPhy transmits the packet to the channel with the initial transmission power.

The channel is simulated by the YansWifiChannel class. YansWifiChannel calculates for a new transmission the transmission delay and the power loss due to the propagation of the transmission for every YansWifiPhy attached to it. The delay is used to schedule a call of the YansWifiChannel reception handler for each YansWifiPhy. When the simulated transmission reaches the wireless radio, the YansWifiChannel reception handler is due. The handler forwards the transmitted packet together with the reception power level to the corresponding YansWifiPhy.

In the YansWifiPhy at the receiving station the power loss due to the transmission is computed. From the degradation of the signal and the modulation given in the
WiFiMode, the SNR and the probability of the reception is computed.

Packet Reception

If a WifiPhy successfully receives a packet it forwards it up to MacLow. If a packet is not received successfully MacLow is notified by WifiPhy. In both cases the measured SNR is forwarded to MacLow. MacLow copies the duration announced in the packets header and updates the DcfManager. In a next step it checks if the packet is addressed to it and determines the packet’s type and parameters. If the packet is addressed to a group, the broadcast address or directly to the WifiNetDevice the MacLow instance belong to, it is processed. If the packet is a RTS packet the MacLow schedules the transmission of an CTS after a SIFS. The CTS is then generated and forwarded directly from MacLow to WifiPhy. If the packet requires an acknowledgment, MacLow schedules the transmission of one after a SIFS delay.

The packet is forwarded up to MacRxMiddle. In MacRxMiddle the detection of duplicated packet and the reconstruction of fragmented packets takes place. MacRxMiddle forwards a resulting packet up to the WifiMac. The disaggregation of A-MPDUs takes place in the WifiMac. The WifiMac then forwards the data packets up to the WifiNetDevice or processes packets that represent management frames. The WifiNetDevice then forwards it up to the upper layer at the node.
Chapter 7

Implementation

7.1 Extensions to the ns-3 Network Simulator

7.1.1 Existing Block Ack Support

As basis for the implementation of MRG Block Ack, we started with the implementation of unicast Block Ack support by Mirko Banchi. He implemented the most parts of the Block Ack support defined in IEEE 802.11n Draft 4.00 that is obsolete in the meantime. His implementation provides support for the Immediate Block Ack variant with Basic and Compressed Block Ack Response frames. The Multi TID and Delayed Block Ack variant are not supported. A description of the legacy unicast can be found in the Background chapter and a description of the various Block Ack variants is given in the chapter on the IEEE 802.11aa amendment. The implementation is part of the development version of ns-3 and will may be in the next release. Unfortunately his implementation is still in an unstable development state. Thus is was necessary to spend more time on the implementation of the extensions needed for the MRG Block Ack.

Banchis implementation adds a BlockAckManager class to every instance of the EdcaTxopN class. This manager controls the transmission and retransmission of data packets for this access category and whether the data packet is transmitted with normal acknowledgment or block acknowledgment. The implementation provides a mechanism that starts sending packets with Block Ack policy, if the number of queued packets in the access category exceed a threshold of queued packages in the EdcaTxopN class. By default the threshold is 0 packets. This turns the Block Ack support off.

1http://codereview.appspot.com/144050/show
Figure 7.1: The Block Ack Manager in ns-3 needs to collaborate with nearly all major classes of the WifiNetDevice.

**Block Ack Agreement**

An agreement is managed in the BlockAckManager on the sending side. The BlockAckManager stores in the agreement the data packets for a possible retransmission. On the receiving side an agreement is managed in MacLow. The MacLow stores the data packets to keep track for the Block Ack response.

Before the transmission of every data packet the following test is executed. If no agreement exists and the number of packets queued in the EdcaTxopN exceeds a predefined static threshold, the setup is initiated. In a successful setup of a Block Ack agreement the frame exchange looks like the one in figure 2.11. The originator of an agreement transmits an ADDBA Request frame to the recipient and creates a pending Block Ack agreement in the BlockAckManager that belongs to the calling EdcaTxopN. The originator blocks until an Ack frame is received or a timeout occurs. The recipient sends an Ack frame after the reception of the ADDBA Request, creates an agreement in MacLow and sends an ADDBA Response frame.
back to the originator. When the originator receives the response it is forwarded up from MacLow to the WifiMac via MacRxMiddle. In the classes derived from WifiMac with support for QoS exists a method to forward the ADDBA Response down again to the originating EdcaTxopN respectively BlockAckManager. The BlockAckManager updates the agreement now from pending to established.

During the setup, when the originator has a pending agreement with the recipient and is waiting for the recipient to transmit a ADDBA Response packet, the BlockAckManager blocks all transmissions of data packets to the recipient. This is done to ensure that data frames using Block Ack are transmitted only when the agreement has been set up successfully. The tear down of a Block Ack agreement and the different cases of error handling are done in an analog way.

The implementation supports that two stations can have independent Block Ack agreements for every user priority. Since the every access category manages two user priorities, this means a station can have up to two agreements per destination in one instance of the BlockAckManager.

**Data Transmission using Block Ack**

If a Block Ack agreement exists with a remote destination for a TID and a data packet is addressed to this destination and has the user priority defined in the TID, it is transmitted without acknowledgment, like the transmission to the broadcast address. After the transmission of every data packet, it is stored by the EdcaTxopN via the BlockAckManager in the according agreement for a possible retransmission. Then the EdcaTxopN initializes a check if a Block Ack Request is needed. The check returns either a Block Ack Request header or null pointer.

If EdcaTxopN gets a Block Ack Request from the BlockAckManager, it queues the Block Ack Request in the front of the queue. So the next time the EdcaTxopN gets access to the channel the Block Ack Request is transmitted before pending data packets.

The test inside the BlockAckManager schedules a Block Ack Request if the following is true. Either the number of transmitted data packets has reached half of the originators buffer size, or all packets stored in BlockAckManager for retransmission and all packets queued for transmission in the EdcaTxopN are transmitted. Or in other words if all packets that are addressed to the recipient and have the agreement TID are transmitted. Banchis implementation does not negotiate the buffer size between originator and recipient. The check described above returns true at the latest if 32 data packets have been transmitted.

The recipient stores the received data packets in the MacLow and forwards them up on the reception of a Block Ack. The current implementation only provide correct feedback in the first Block Ack. If this response is lost and requested again, the new Block Ack claims that no data frame has been received. This happens since
the data structure for caching the information which data frames have been seen is missing in the current version.

**Block Ack Request and Block Ack**

The ns-3 Block Ack implementation does only support immediate Block Ack. This means that after the reception of a Block Ack Request the Block Ack is sent after a SIFS delay. The starting sequence number for which the acknowledgment is requested is set in the Block Ack Request by the BlockAckManager. The MacLow of the originator sets the duration of the Block Ack Request so that it covers the complete channel holding time of Block Ack Request and Block Ack.

The Block Ack Request is processed in the MacLow of the recipient. After the MacLow received a Block Ack Request it schedules the transmission of a Block Ack. The MacLow checks for the stored data packets whether they are older than the starting sequence number or in the window of 64 MPDUs defined by the starting sequence number and the size of the Block Ack Information field. These data packets are forwarded up to MacRxMiddle. If they are in the window, the corresponding bit in the Block Ack Information field is set to 1, otherwise to 0.

When the originator receives the Block Ack it is forwarded up from MacLow to the WifiMac via MacRxMiddle. In the classes derived from WifiMac with support for QoS exists a method to forward the ADDBA Response down again to the originating EdcaTxopN and its BlockAckManager. In the BlockAckManager the list of transmitted data packets is checked for packets that exceed the maximal packet lifetime limit. All older packets are removed. For the remaining packets is checked if they are marked as received in the Block Ack. If not they are queued for a retransmission.

### 7.1.2 IGMP Replacement

The IEEE 802.11aa amendment suggests to use IGMP snooping at the access point, to learn the group memberships of associated stations. Since ns-3 does not provide support for IGMP, this can not be implement in that way. Instead a workaround is provided.

The class MrgMappingContainer takes tuples of groupcast MAC address and NetDevice. The device is the WifiNetDevice of the nodes that represents the wireless station. For each membership the container takes one tuple. The helper wrapper for the WifiNetDevice is extended, so that it can take the MrgMappingContainer as additional parameter. Due to this the MrgMappingContainer must be created and filled with memberships after the creation of the nodes WifiNetDevices and before the creation of the access point and its WifiNetDevice. The helper wrapper sets a pointer to the MrgMappingContainer in the WifiRemoteStationManager at the access point.
Derived from the WifiRemoteStationManager base class, some classes were created to provide support for MRG traffic, like the AarfWifiManager. On the creation of an instance of AarfMrgWifiGroup, the AarfWifiManager copy the MAC addresses of the stations into it. This setup cannot be done using a static script. The WifiRemoteStationManager creates the instances of WifiRemoteStation, the first time it has to access the instance for a specific destination. The MAC address is assigned to the NetDevice within the setup process and not visible to the helper wrapper. Because of this the MrgMappingContainer takes the NetDevice and get the MAC address at run-time.

7.1.3 Association Identifier (AID) Transmission

The association identifier (AID) is a unique number, negotiated between access point and non-AP station during the association process. It is a number in the range of 1 to 2007 and it is stored in a field of size 2 bytes.

ns-3 already provides management frames for association request and response that contain an AID field. But ns-3 does not exchange AID by default. The AID is only needed in some protocols within 802.11. The decision which AID is assigned to a new station is not part of the standard. For the MRG Block Ack it is optimal when the range of assigned AIDs is compact. Compact means that between two assigned AIDs exist as few as possible non assigned AIDs.

The QapWifiMac, QstaWifiMac and WifiRemoteStationManager has been extended to support AID negotiation. In the association phase, the access point sends the AID with the association response to the non-AP station. The AID generation is done in the WifiRemoteStationManager at the access point. It maintains a counter modulo 2007. The AID assigned to this station is stored in the corresponding WifiRemoteStation, at the access point. As soon as a frame is transmitted to the non-AP station, an instance of WifiRemoteStation is created. This instance will not be deleted before ns-3 terminates. This ensures that a station gets the same AID again, if it is disconnected and needs a re-association.

In the non-AP station the AID is stored in the QstaWifiMac.

7.1.4 Extensions to given Components

It has been necessary to make changes in nearly every class the WifiNetDevice consists of. Most of the changes are needed to provide support for the new frame types MRG Block Ack introduces. Due to the class design that represents the frames, it is necessary to provide a new handler for every new frame type in every class that processes this new frame type. Only for some limited cases a group of frame types can be handled by one handler. Next the components, that needed changes to implement more functionality for, are presented.
QapWifiMac

QoS data frames contain a sequence control field. This field contains a 12 bit sequence number field and a 4 bit fragment number field. The fragment number is of no interest for us, since group addressed traffic shall not be fragmented according to the IEEE 802.11 standard. The legacy IEEE 802.11 defines that an originator must maintain a separate sequence number counter per destination. For all group addressed and broadcast traffic the use of a common sequence number counter is defined.

Due to backward compatibility this is not changed by the IEEE 802.11aa. The Block Ack Request and Block Ack frames with a starting sequence number and a bit-field, are designed for traffic in bursts with continuous sequence numbering. This means for MRG Block Ack that the access point must reorder the arriving traffic and forward it in bursts to the group. This queuing and reordering is not part of the standard.

A simple multiplexing mechanism to buffer incoming traffic addressed to different groups at the access point and forward it in bursts of frames with a common destination has been designed. The traffic is stored in separate queues per tuple destination and TID. To implement a variant of fair queuing, the multiplexing chooses the queue with the most packets for transmission. If the queue has collected the suggested number of data packets per block and the EdcaTxopN is done with the transmission and possible retransmission of older packets, the packets from the queue are forwarded to the EdcaTxopN corresponding to the TID. The number of forwarded packets at once is limited by the suggested block size. The suggested block size is determined by BlockAckManager in the EdcaTxopN.

This mechanism is sufficient for our scenario. We are only interested in streams with a constant date rate and a constant packet size which are addressed to multicast groups. For our tests we only generate streams for transmission using the access category Video. Thus we do not have to take streams with a variable data rates into account or the different priorities they might have.

Wifi Remote Station Manager

The WifiRemoteStationManager is an abstract base class and implements the link adaption in the ns-3 wifi devices. As soon as a frame is transmitted to or received from a new WifiNetDevice, in the WifiRemoteStationManager a new instance of WifiRemoteStation is created. The WifiRemoteStationManager controls the setup and reset of a WifiRemoteStation on association and disassociation. The WifiRemoteStation stores the set of WifiModes supported locally and by the remote station. It is in the responsibility of the WifiMac to copy all WifiModes for WifiPhy into it. To provide support for MRG traffic, the WifiRemoteStationManager and the WifiRemoteStation has been extended. Most of the extensions to the manager are accessor methods to realize the access to the MrgMappingContainer that
7.1. EXTENSIONS TO THE NS-3 NETWORK SIMULATOR

replaces the IGMP support.

The creation of new WifiRemoteStations is done on the fly in the manager when it is looked up the first time using the address of the corresponding destination. The manager has been extended so that for a group address a special class that manages the link adaptation for the group is created. This class is derived from WifiRemoteStation to keep the needed changes in the rest of the WifiNetDevice minimal. It is only implemented in the classes derived from WifiRemoteStationManager that provide support for MRG Block Ack, like IdealMrgWifiManager. In this case an instance of IdealMrgWifiGroup exists in the IdealMrgWifiManager belonging to the WifiNetDevice of the access point. On creation the IdealMrgWifiGroup also creates a WifiRemoteStation for every station that is member of its corresponding group.

The abstract base class WifiRemoteStationManager only contains empty implementations of the virtual methods needed for MRG Block Ack support. Only the derived classes with support for MRG Block Ack contain the actual implementation.

The basic WifiRemoteStation to support the MRG Block Ack has been extended. This extensions are for use in the instances contained in the WifiRemoteStationManager at the access point. The first extension is a data structure to keep track of the multicast groups, the station is member of, and for which an MRG Block Ack agreement exists. The second extension is for the communication with the WifiRemoteStation that represents the group destination. Since the different data rates are represented in instances of the class WifiMode, no order exists.

When the transmission of a group address packet starts, the MacLow looks up the WifiRemoteStation for the destination, in order to determine the data rate to use for the transmission. The special WifiRemoteStation for a group address will not determine the data rate by itself. Instead it asks the WifiRemoteStations, for the remote stations that are member of the group and return the minimum. Here the minimum is the fastest and least robust data rate so that all stations should be able to receive the transmission correctly.

On the reception of a transmission the WifiRemoteStation corresponding to the originator of the transmission is informed about the measured SNR. In the case of a MRG Block Ack, the originator is not the group but the station in the group. So the WifiRemoteStation corresponding to this group member gets updated.

The advantage of this approach is that not only feedback from the MRG Block Ack is used for the link adaptation, but also the feedback from other transmissions. This is the case because it exists one WifiRemoteStation per destination. So the station polled as a member of a group is also polled for unicast traffic.
EdcaTxopN

The EdcaTxopN works together with the MacLow to implement the synchronous unicast transmission of MPDUs. The DcfManager calls the EdcaTxopN, once it gains access to the channel. The EdcaTxopN then forwards a packet to the MacLow. Together with the packet a signal handler and a set of transmission parameters is forwarded. The MacLow forwards the packet to WifiPhy. When the Ack packet arrives in time, the MacLow uses the handler to signal the successful transmission to the EdcaTxopN. Otherwise the failure or cancel of the transmission is signaled. The EdcaTxopN blocks and does not request further channel access from the DcfManager until the signal from the current transmission arrives.

The class with the signal handler belongs to the EdcaTxopN. It has been extended to support the reception of multiple packets in response to an initial transmission. The MRG Block Ack Request frame expects none or multiple MRG Block Ack frames in response. With a theoretical limit of 2007 responses, if all AIDs are addressed. To support the reception of multiple MRG Block Ack responses, in the transmission parameters for the MRG Block Ack Request the EdcaTxopN sets the number of expected responses. The MacLow signals the reception of a MRG Block Ack, the reception of all outstanding MRG Block Ack responses or the missing of outstanding responses to the EdcaTxopN.

In the EdcaTxopN a data structure has been implemented to keep track of the MRG Block Ack responses. The MPDUs feedback is polled for, shall be retransmit until all stations acknowledge their reception or their lifetime limit expires. The access point shall retransmit a MRG Block Ack Request, naming only the AIDs of the stations has received no MRG Block Ack in response for the last MRG Block Ack Request. Together with the signal for missing MRG Block Ack response, the MacLow forwards a list of those stations no MRG Block Ack has been received from. The EdcaTxopN now schedules a MRG Block Ack Request for transmission as soon as it gains channel access again, naming exactly these stations to send a MRG Block Ack in response.

The MacLow signals the reception of the last MRG Block Ack if all outstanding responses are received. The EdcaTxopN now creates a Block Ack with the information aggregated from the responses. In the Block Ack a packet is marked as received, if all stations have marked it as received. This Block Ack is forwarded to the BlockAckManager as the Block Ack from the group. The BlockAckManager now schedules packets not received by at least one station for retransmission to the group address.

The EdcaTxopN supports the transmission of aggregated MSDUs into one MPDU for unicast traffic. The so called A-MSDU. An A-MSDU consists of multiple MSDUs aggregated into one MPDU. A-MSDU is defined in IEEE 802.11n. The EdcaTxopN has been extended with support for group addressed traffic using the MRG Block Ack policy. The support has been implemented by using the given code. The EdcaTxopN peeks the next packet in its queue and aggregate it to the current, as
long as the resulting A-MSDU size is still under the predefined limit. The resulting A-MSDU is then transmitted like a normal packet. It is important to mention here, that the MSDUs in an A-MSDU stay aggregated until the disaggregation in the recipients WifiMac takes place. This implies also that a A-MSDU is retransmitted as one packet in ns-3, even if it would be probably better to transmit the original MSDUs separated.

This behavior is due to the strict synchronous design of the EdcaTxopN class. Despite its name it supports only the transmission of one frame per TXOP. Thus it has originally not been not designed to provide support for the retransmission of multiple outstanding MSDUs. The only support for asynchronous transmission of multiple MSDUs and the management of retransmissions in the case of errors is the Block Ack mechanism implemented in the BlockAckManager.

**MacLow**

The MacLow also needed some extensions to provide support for the MRG Block Ack policy. The class to forward transmission parameters together with the packet has been extended. For MRG Block Ack some additional flags and most important the list of stations addressed in the MRG Block Ack Request were needed.

In MacLow the calculation of the channel holding time takes place. The time for the actual transmission is calculated via a method in WifiPhy. This method is passed the complete size of the packet to send and the WifiMode. For the legacy control frames like Ack or RTS and CTS frame the duration is a constant with respect to the basic data rate. Thus the duration of this frames is computed once during the setup. The Block Ack and MRG Block Ack frames are control frames. But contrary to the legacy control frames they are transmitted using the actual data rate and not only the basic data rate. Thus the duration may differ due to different data rates. The situation is even more complex for MRG Block Ack Request frames. The size of this frame is variable with respect to the addressed stations in the MRG BAR Information field.

The given implementation only supports as an interim solution a hard coded duration for Block Ack frames. One value for Compressed Block Ack and one for Basic Block Ack. These values are optimized for the use with a parametrization for IEEE 802.11a. The MacLow has been extended so that now all durations are calculated with respect to the size of the packet and the used WifiMode.

To support the caching and buffering of received MPDUs at the stations that are member of at least one multicast group the MacLow has been extended in the following way. The MacLow may not forward up MSDUs in the wrong order. This is easy to ensure for the normal Ack policy. Since only at most one MSDU is in transit at a time. When the MSDU is fragmented into multiple MPDUs, the MacLow forwards up every received MPDU to the MacRxMiddle. The reconstruction and reordering of a fragmented MSDU is implemented in MacRxMiddle.
CHAPTER 7. IMPLEMENTATION

To prevent reordering in asynchronous transmissions with MRG Block Ack policy the MacLow provides a buffer that provides space for up to $N$ MPDUs. Where $N$ is the buffer size from the negotiation of the MRG Block Ack agreement. The buffer stores all MPDUs that are not received in order. As soon as all MPDUs of a MSDU are received and no older fragmented MSDU is present in the buffer, the MSDU is forwarded up and removed from the buffer. This is done to minimize the delay and jitter caused by the buffering. In an ideal case, the buffer is always empty. Because of this the buffer does not provide the information needed for feedback.

To provide feedback, the MacLow maintains a cache. According to the IEEE 802.11 this cache takes for unicast traffic using Block Ack tuples of the source address, the TID and MPDUs sequence number. The cache stores the latest $N$ tuples for every received MPDU with a source address and TID a Block Ack agreement exists for. For group address traffic the cache stores tuples containing the destination address, the TID and the MPDUs sequence number. The cache stores the latest $N$ tuples for every received MPDU with a destination address and TID a MRG Block Ack agreement exists for.

While the buffer data structure was already present, the cache data structure was missing. The cache data structure and the methods to maintain the buffer and cache has been implemented. The methods in MacLow has been extended with support only for group addressed MRG Block Ack, but not for unicast Block Ack. In the current implementation the originator and the recipient always agree on the maximal buffer and cache size $N$ of 128 MPDUs.

7.2 SNR-based Link Adaptation

With the change from the synchronous transmission using the legacy acknowledgment policy to the asynchronous Block Ack policy for unicast or the MRG Block Ack policy for group addressed traffic the given mechanisms for link adaption do not work. This is due to the fact that not after every packet feedback is provided. Using legacy Ack policy after every transmitted data frame, an Ack frame transmitted in response signals the reception. Its absence indicates the loss of the transmission. The most of the given link adaption mechanisms are designed upon this feedback.

The problem consists of two separate problems. The first is to determine the physical data rate to transmit the next block of data frames. The second is to determine how often to request a Block Ack for feedback, or in other words how many data frames should be transmitted in one block? The IEEE 802.11aa does not define any link adaptation or block size adaptation algorithm.

A link adaptation has been implemented using the measured SNR for the last received MRG Block Ack in ns-3. The size of a block is determined from the length of the data frames transmitted with success up to the first error in the last frame. The idea behind this approach is to determine from the measured SNR the data
rate for the next transmissions and to adapt the size of the block to the time the wireless channel does not change its conditions.

Our approach ignores a problem that occurs with multiple streams. When the streams are addressed to different groups and not all stations are member of all groups, the feedback provided from the MRG Block Ack is a measurement of the channel conditions for a group of remote stations. The next block scheduled can be for another group of remote stations with other channel conditions. The conditions may simply differ, due to the different distances the different stations have to the access point. Figure 7.2 illustrates this for two group addressed streams.

To overcome this problem, our link adaptation does not only take the measured SNR from the MRG Block Ack frames into account but also other traffic from this stations. So the data rate for the next transmitted block does not only depend on the feedback from the last group addressed traffic. This also reflects real world behavior where application layer feedback is transmitted from the station via the access point.

A more demanding solution to this problem may be subject for further development. An implementation and testing with ns-3 would require massive changes in the WifiNetDevice.

### 7.2.1 Link Adaptation

For the link adaptation the given IdealWifiManager has been taken as basis. This manager implements a rate control algorithm similar to the RBAR algorithm to determine an ‘ideal’ data rate for the next transmission.

On the reception of a packet the MacLow gains the measured SNR from the WifiPhy. This SNR is forwarded from the MacLow to the WifiRemoteStation corresponding to the origin of the transmission.

The IdealWifiManager generates at initialization a table matching SNR values to appropriate data rates. For this the IdealWifiManager has an extra pointer to the WifiPhy. For every WifiMode provided by the WifiPhy and a fixed maximal acceptable bit error rate (BER) the IdealWifiManager determines a SNR value as threshold. The BER defaults to $10^{-6}$. The calculation is done via the error model.
provided in the YansWifiPhy. YansWifiPhy calculates the SNR of a transmission from the used WifiMode and the determined BER of that transmission. Doing so the thresholds in the table are generated with respect to the parametrization of the wireless channel model in the YansWifiPhy.

The IdealWifiRemoteStation now determines the maximal date rate using the last SNR value signaled by the MacLow and this precomputed table and returns the corresponding WifiMode.

The major extension in the IdealMrgWifiManager is the IdealMrgWifiGroup. This class is derived from the WifiRemoteStation and represents the group as a remote destination. Every time this class is requested to return a WifiMode for the next group addressed transmission, it requests for all remote stations that are member of the group the data rate for a direct transmission to this member. After this it calculates the minimum of the possible data rates and returns this minimum.

To implement the request of a data rate between instances of WifiRemoteStation in one WifiRemoteStationManager the WifiRemoteStationManager has been extended with virtual methods not to return the corresponding WifiMode but instead the index of the WifiMode with respect to the order in the internal data structure. The actual implementation of these methods then takes place in IdealMrgWifiRemoteStation. Figure 7.3 illustrates the approach. This is necessary because of the fact that the WifiMode class describes a data rate with all parameters but does not provide an order among them.

A benefit from this approach is that not only the measured SNR from the last received MRG Block Ack is taken into account for the data rate for the next group addressed transmission, but also other traffic received more recently from a station that is member of this group.
7.2. SNR-BASED LINK ADAPTATION

7.2.2 Block-Size Adaptation

To provide the needed information for the adaptation of the block size the Block-AckManager has been extended. The manager keeps track of each agreements parameters and packets with outstanding acknowledgments. The data structure has been extended to keep track of the additional parameter block size.

The EdcaTxopN stores a copy of each packet transmitted using Block Ack policy in the corresponding agreement in the BlockAckManager. When the EdcaTxopN gains access to the channel, it first transmits the packets queued for a retransmission by the BlockAckManager. On the reception of a Block Ack frame, the manager checks for each outstanding packet if its successful transmission is acknowledged. If not, the packet is copied from the agreement to a retry queue. If a packet is acknowledged as successful transmitted, it is removed from the agreement.

The BlockAckManager has been extended to determine the number of data packets to be transmitted in the next block. The measured value is the number of packets queued for retransmission after one Block Ack. This is the number of transmitted packets that has not been received by the recipients. With the number of errors in one block the size of the next block is determined. If errors occur the block size is set to an initial value of 2 packets. If no error occur the block size is increased by 2.

In addition to this a more aggressive approaches with multiplicative increase has been tried. But the only result that has been achieved was a higher error rate. The bad performance is probably due to the missing of a protective mechanism for the transmission of the block of data packets. A block of data MPDUs under Block Ack policy should be protected from other transmissions. This could be done for unicast transmissions with an RTS/CTS exchange before the block of data MPDUs that announces the transmission duration of the complete block. For unicast and group addressed transmissions the duration of the complete block can be announced by setting the duration of the first MPDU in the block to cover the complete block. This works because of the limitation that all MPDUs in the block should be addressed to the same destination. The size of such a protected block should not exceed the maximal TXOP duration, announced by the access point.

Since ns-3 does not provide support for multiple packets in one TXOP and the implementation of the CSMA/CA mechanism does not allow to tweak the transmission duration field of a package without side effects, it has not been possible to implement a protective mechanism for blocks of MPDUs transmitted under Block Ack policy.
7.3 More Reliable Groupcast Block Ack

The MRG Block Ack has been implemented using the given Block Ack support as start and extended the WifiNetDevice and its components with the needed functionality. For the forwarding and processing of the different new frames the given methods for Block Ack have been extended to support also MRG Block Ack. The given data structures have not been altered. To support MRG Block Ack, new data structures have been added to multiple classes. The WifiRemoteStation has been extended to maintain the group membership at the access point. The MacLow has been extended with a caching data structure to keep track of the received MPDUs.

7.3.1 MRG Block Ack Agreement

The 802.11aa defines the extended ADDBA Request and Response frames. these have been implemented in ns-3, using the given ADDBA frames. The main difference is the presence of an additional address field for the group address.

The setup process is initialized in the QapWifiMac after the completion of the association with a station. To complete the association, the QapWifiMac generates an association response. The non-AP station acknowledges the reception of this response with an Ack frame. The reception of this acknowledgment is signaled to QapWifiMac. For every multicast group the new associated station is member of, the setup of a MRG Block Ack agreement is initialized. The QapWifiMac calls the EdcaTxopN for a hard-coded set of TIDs. EdcaTxopN first calls the WifiRemoteStation for the group. This will bootstrap the WifiRemoteStations and will ensure that all WifiRemoteStations are generated and the needed information about the group membership are copied into them from the MrgMappingContainer.

The EdcaTxopN now generates an Extended ADDBA Request frame that is addressed to the non-AP station and contains the group address and TID the agreement is for, queues it, and transmits the frame as soon as it gains access to the channel again. During the frames creation, in the BlockAckManager an OriginatorBlockAckAgreement is created in state pending. To prevent traffic from being transmitted using MRG Block Ack while the agreement negotiation takes place, the access point also blocks all other traffic addressed to the group.

When the Extended ADDBA Request is forwarded up to the non-AP stations QstaWifiMac a Extended ADDBA Response is created and and queued for transmission in the EdcaTxopN corresponding to the TID from the request. At the same time a BlockAckAgreement is created in MacLow to store received data packets for reordering. In addition to this a cache is created in MacLow to keep track of the last received data packets sequence numbers for a MRG Block Ack agreement.

The Extended ADDBA Response is forwarded up to the QapWifiMac. From there it is forwarded down to the EdcaTxopN respectively the BlockAckManager corre-
7.3. MORE RELIABLE GROUPCAST BLOCK ACK

sponding to the TID from the response. In the BlockAckManager the agreement is updated. The status is set from pending to established and the transmission parameters are updated if necessary.

A mechanism to tear down a MRG Block Ack agreement explicitly by sending a DELBA frame with a MRG Group Address present has not been implemented. This is done because of the lack of dynamic group membership in ns-3. The access point keeps track about the established MRG Block Ack agreements with a non-AP station in the corresponding WifiRemoteStation. In the case of a disassociation, the agreements are invalid and deleted. After a re-association the agreements are created again.

7.3.2 Data Transmission using MRG Block Ack

The transmission of data packets using MRG Block Ack policy equals the transmission using the Block Ack policy. If an agreement for a group and a TID exists, transmissions addressed to this are stored in the BlockAckManager for a possible retransmission.

After each transmission the BlockAckManager checks if a request must be scheduled. If a request is needed and the remote destination is a group, the EdcaTxopN takes care to schedule a MRG Block Ack Request instead of an Block Ack Request packet. This is done by queuing the MRG Block Ack Request for transmission before all other pending packets. The test checks two conditions. The first one tests if the number of transmitted data frames since the last MRG Block Ack Request exceeds the block size suggested by the BlockAckManager. The suggestion is based on the given feedback as described above. The second one tests if all packets in the EdcaTxopN for transmission and in the BlockAckManager for retransmission have been transmitted. If one of the two conditions holds true, a MRG Block Ack Request is queued for transmission.

If all pending data packets in the EdcaTxopN and BlockAckManager have been transmitted, the queuing mechanism in QapWifiMac forwards the next block of data packets to the EdcaTxopN as described above. The queuing mechanism ensures that only blocks of data packets addressed to the same group and with consecutive sequence numbers are transmitted as needed by the MRG Block Ack.

7.3.3 MRG Block Ack Request and MRG Block Ack

The initial MRG Block Ack Request is generated in the BlockAckManager. Figure 7.4 illustrates the process. The MRG Block Ack Request is addressed to the group and extended with the MRG BAR Information field. The BlockAckManager requests the MRG BAR Information field, from the WifiRemoteStation that controls the data rate for traffic addressed to that group. The field contains the group
members that should reply to the request. Our implementation asks all members for feedback.

![Diagram](attachment:image.png)

**Figure 7.4:** The diagram shows the generation of a MRG Block Ack Request. The MRG BAR Information field with the addressed stations is generated by the WifiRemoteStation that controls the traffic to that group address.

The MRG Block Ack Request is now transmitted like the unicast Block Ack Request. The difference is in the transmission parameters and the event handler. The EdcaTxopN also sets the list of stations that are expected to reply. A station a MRG Block Ack is received from, is removed from the list of expected responses in the MacLow. The MRG Block Ack is forwarded up to the originating EdcaTxopN via the event handler. If all expected responses are received the corresponding event handler signals this to the EdcaTxopN. If the MacLow notifies the missing of expected MRG Block Ack packets, it forwards the list of stations with still outstanding responses back up to the EdcaTxopN.

In the EdcaTxopN, a new MRG Block Ack Request is generated and scheduled for transmission as soon as the EdcaTxopN gains again access to the channel. The stations addressed in the MRG BAR Information field of the new MRG Block Ack
7.3. MORE RELIABLE GROUPCAST BLOCK ACK

Request, are exactly those with still outstanding MRG Block Ack as specified by IEEE 802.11aa.

Since ns-3 provides no support for IGMP and multicast group membership is static, in the WifiRemoteStation only those stations are added to the initial MRG BAR Information field an agreement exists for. An agreement can only exist if a station is associated.

Figure 7.5: Sequence diagram shows an access point that receives an MRG Block Ack frame.

The received MRG Block Ack responses are processed in the EdcaTxopN that originated the MRG Block Ack Request. Until the MacLow signals that all expected responses are received, the BA Information field is stored in a data structure. If the feedback from all stations is present, the EdcaTxopN generates a Block Ack packet and forwards this to the BlockAckManager. The Block Ack is addressed in a way to be processed by the BlockAckManager as the response from the group address. In the Block Ack the BA Information field is set using the stored information from the MRG Block Ack responses. At this a data packet is marked as received, if and only if it was marked as received in all the MRG Block Ack responses. This will cause the BlockAckManager to schedule all data packets that with not expired lifetime,
that were not received by all stations for a retransmission, as specified by IEEE 802.11aa.

Figure 7.6: The diagram shows a non-AP station that responds to an MRG Block Ack Request with a MRG Block Ack frame.

In the non-AP station the MRG Block Ack Request is processed in the MacLow. The MacLow maintains a cache as described before with information about the last received data packets subject to an MRG Block Ack agreement. If a MRG Block Ack Request is received, the MacLow generates a MRG Block Ack as response. The BA Information field is set according to the information stored in the cache and the MRG Block Ack is transmitted with a delay after the end of the reception of the request. The delay duration is determined by the AID of the non-AP station, the other AIDs mentioned in the MRG BAR Information field and the data rate the MRG Block Ack Request was transmitted with.
Chapter 8

Experiments

In this chapter the results obtained with our implementation of the MRG Block Ack in ns-3 are presented. A common simulation environment for all scenarios has been defined. Then the behavior of the MRG Block Ack in different scenarios has been tested. The main audio/visual application in the scenario are IPTV streams from a media server to several clients. This means the main traffic flow on the wireless network is unidirectional from the access point to the wireless stations.

8.1 Simulation Setup

The simulation focus on one access point and a limited number of wireless stations. This is most common for most small office and media at home scenarios. Further a wired-cum-wireless topology with a dedicated media server that provides the streams has been assumed.

8.1.1 Network Topology

Figure 8.1 shows the topology of the simulated network. The network consists of a wired network and a wireless network. The two networks are connected by the access point. All stations in the network are represented by nodes. A node is equipped with NetDevices, a protocol stack and applications as illustrated in figure 6.2. The nodes of the different networks are organized in NodeContainer. Three different NodeContainers are defined. The CsmaNodes for the wired nodes, the StaNodes for the wireless non-AP nodes, and the APNode container for the access point. The node for the access point is also member of the two other containers. The different network devices are also stored in containers. The devices in a container usually simulates network devices connected to a common channel with a common collision domain.
The wired network is simulated by the ns-3 CSMA Device model. This model simulates a simple wired bus model. The wired network devices are stored in a NetDeviceContainer. Three different NetDeviceContainers are defined. The CsmaDevices for the wired CsmaNetDevices, the StaDevices for the wireless WifiNetDevices that are parametrized to simulate wireless QoS non-AP stations and the APDevice for the wireless WifiNetDevice that is parametrized to simulate a wireless QoS access point. The wired channel is parametrized to provide 100Mbit/s throughput. The configuration of the wireless channel is more complex and is described in the next section.

![Network Topology Diagram](image)

Figure 8.1: The figure shows the network topology used for our experiments. It contains a wired network connecting the traffic sources at node Csma_1 with the access point at node Ap. At the nodes in the wireless network received traffic is monitored.

### 8.1.2 Wireless Channel

For the wireless channel a Rayleigh fading model has been assumed. Due to the obstacles in the assumed home or office environment, a multi-path propagation with no dominant line-of-sight path is most probable. The wireless channel is simulated by a YansWifiChannel with an NakagamiPropagationLossModel. This model
is parametrized so that it models Rayleigh fading. The propagation delay and
the error rate are computed by the default ConstantSpeedPropagationDelayModel
and YansErrorRateModel. The link adaptation is provided by the IdealMrgWifiRe-
modelStation. For the different experiments different setups using the nobility model
in ns-3 have been provided.

The WifiNetDevice is, unlike other network devices provided by ns-3 that are at-
tached to a channel by being in a container, attached to the wireless channel in the
following way. First the different components are instantiated. Then the device is
created with this components. At first a YansWifiChannel is created. This simula-
tion of a wireless channel can be parametrized with propagation loss models. Then
the YansWifiPhy is created. The WifiPhy simulates the wireless radio. The Wifi-
Phy takes the wireless channel it is using as a parameter. Independent from this the
WifiRemoteStationManager is instantiated and parametrized. The same holds for
the WifiMac. The station manager determines the strategy for link adaptation and
the WifiMac determines the type of the wireless station. The WifiNetDevice is the
skeleton these components are attached to. In addition, the node a WifiNetDevice
is connected to must be part of a mobility model. This is needed to determine
the distance between two WifiNetDevices and the degradation of the transmitted
signal.

8.1.3 Traffic Setup

Two streams with a constant data rate of $4\text{Mbit/s}$ have been set up. The second
stream starts $2.5\text{seconds}$ after the first. The streams are transmitted with user
priority set to 4, so both streams are processed in the video access category at the
access point. The two streams simulate the traffic load caused by two IPTV streams.
The ns-3 does not provide an implementation of the RTP protocol defined in RFC
3550 [26] used for audio/video streaming applications. In order to compensate for
this, the tag system provided by ns-3 has been used. Tags can be attached to a
packet at any time. These tags travel with the packet through the different layers
and networks but are not counted in terms of packet size and traffic load. In ns-3 a
packet can not be corrupted. It can only be transmitted successfully or be lost. If it
is lost, all attached tags are lost too. The traffic source has been extended to attach
a tag with a sequence number and a tag with the creation time of the packet. We
use this information in the packet sinks to determine the throughput, delay, jitter
and packet loss rate. The definition of this variables is given in the second half of
this section.

An IP stack is added to each node in the simulation. The nodes in the wired
network are assigned IP addresses from the 10.1.1.0 subnet. The nodes in the
wireless network to addresses from the 10.1.2.0 subnet. In both cases the netmask is
255.255.255.0. The streams are addressed to the two multicast groups 225.1.2.4 and
225.1.2.5. They are referred to as stream A and stream B. The APNode implements
dynamic routing for the unicast traffic by default. For the multicast traffic static
routes at the APNode and a default route at the node with the traffic source have to be set up. The setup is illustrated in figure 8.1. The group membership of a wireless non-AP station is expressed by the presence of a packet sink for that group at the corresponding node. Due to the lack of support for IGMP in ns-3 the APNode can not learn about the multicast group membership by IGMP snooping as suggested by the IEEE 802.11aa. Instead a MrgMappingContainer that is attached to the WifiNetDevice provides the needed information. The container contains a tuple consisting of multicast IP address and NetDevice for every NetDevice that is listening for the transmissions address to this multicast IP address.

The traffic sources are instances of the OnOff application provided with ns-3. The application can be parametrized to generate traffic stream with an average data rate. The application name refers to the alternating On and Off states. The application can be parametrized to stay in each state for a period of time. This results in a traffic burst in the On state. The duration of the Off state is the time between two traffic bursts. We parametrize the application to produce a stream of UDP packets with a payload of 1328\textit{Bytes} at a constant data rate of 4\textit{MBits/s}. The payload size is chosen because in an IPTV stream a UDP packet contains seven MPEG packets with a size of 188\textit{Bytes} plus a RTP header of 12\textit{Bytes}. To each of these packets we add tags with a sequence number and creation time as described before.

The measurement of the achieved throughput, delay, jitter and packet loss rate is done at the Application Layer because we are interested in the influence the MRG Block Ack has on the applications that use it. These are audio/video streaming applications that have strict requirements to these parameters and are highly affected by violations of these requirements. The throughput is measured by summing up the payload size of every received packet in a sampling interval. The actual throughput per sampling interval is then computed with the formula

$$
\frac{1}{\text{sampling interval}} \times \frac{\text{cumulative payload} \times 8}{10^6}
$$

which returns the throughput in \textit{Mbit/s}. The sampling interval is set to 0.1\textit{seconds}. For the computation of the loss rate the packets are not evaluated per sampling interval, but instead the packet sequence numbers are evaluated over the complete simulation duration. This is done because the small number of packets per sample provides only a crude granularity. If a packet with a consecutive sequence number is received in the application, the number of received packets is increased. Otherwise the number of missed packets is derived from the difference between the consecutive seen sequence numbers. The loss rate is then expressed as fraction of the sum of received and the sum of missed packets

$$
\frac{\text{received packets}}{\text{missed packets}}
$$

The delay and jitter introduced by the network between the traffic source application
and the sink application is calculated according to the definition given in the RTP protocol definition in RFC 3550 [26]. The delay is the difference between the time stored in the tag attached to the packet and the current time. The jitter is computed by the formula

\[ jitter = jitter + \frac{1}{16} \cdot (|\delta| - jitter) \]

with

\[ \delta = (\text{current packet receive time} - \text{previous packet receive time}) - (\text{current packet transmission time} - \text{previous packet transmission time}) \]

The measured jitter is updated every received packet. The value returned every sampling interval is the last updated jitter.

In addition to this measuring at the Application Layer ns-3 provides the output of trace files in the pcap format. This common format for traces of network traffic can be read by tools for network analysis like wireshark[1]. The trace file is created per NetDevice and contains all packets transmitted or received by the device. wireshark is used to analyze the changing channel usage introduced by the MRG Block Ack to compensate the changing channel conditions. The Block Ack mechanism is an automated repeat query (ARQ) mechanism that retransmits packets to compensate from losses.

8.2 Tests Using MRG Block Ack

In this section the simulation results using our implementation of MRG Block Ack and our SNR-based link adaptation are presented. First the achieved improvements compared to the given solution are presented. In the next step it is examined how a single station with reception problems degrade the complete system. Then different approaches to deal with such a situation are tested.

8.2.1 Achieved Improvement

This experiment shows how the use of MRG Block Ack and SNR-based link adaptation improves the transmission of group addressed traffic.

For the test the wireless non-AP station are placed in a line with increasing distance to the access point as illustrated in figure 8.2. The ns-3 mobility model is dimensionless. Thus the stations are 7 units away from each other but the complete system

Figure 8.2: Placing the stations in a line with increasing distance to the access point, for the tests comparing the legacy IEEE 802.11 with our implementation of the IEEE 802.11aa MRG Block Ack.

is parametrized by default in a way that one unit is approximately one meter. The group membership of the three stations is as described in the simulation setup. The stations Wifi_1 and Wifi_3 are receiving Stream A and Stream B. The station Wifi_2 is receiving only Stream B.

Figure 8.3: Comparison of achieved throughput without MRG Block Ack. Left: legacy implementation of ns-3. Right: our queue giving each stream equal channel access. Top: throughput achieved at the application layer. Bottom: throughput at the link layer.

The MRG Block Ack mechanism and the needed queuing transform the stream from a constant packet rate into a transmission with bursts of packets. This results in a sawtooth curve in the diagram. The results as seen by the application are presented as smoothed curves to enhance visibility. For the station Wifi_2 the raw measured results are presented. The results as seen at the link layer are also presented as smoothed curve. From this smoothed curve the average throughput at the link layer is computed and presented in a bold line. In figure 8.3 the achieved throughput is illustrated. Both streams are transmitted using the slowest and most robust physical data rate with 6 Mbps. In this situation the streams would need
more time than available to transmit all data over the wireless channel. The result is packet loss due to an over-saturated wireless channel. In addition the impact of our implementation of fair queuing is presented. The available channel capacity is fairly distributed between the two streams as the figure 8.3 shows.

If our implementation of MRG Block Ack with SNR-based link adaptation is applied on the given setup the traffic changes as illustrated in figure 8.5 on page 112. The SNR-based link adaptation determines the physical data rate with 24 Mbps in order to enable the farthest station to receive. The delay and jitter introduced by the queues and the retransmission rounds are negligible small. The loss rate is about 1.5%. Unfortunately it is not possible to produce statistics about the average block size or the number of lost data frames per block with the implementation. These statistics would be interesting because of the direct relation between the number of data packets in one block and the overhead produced by the MRG Block Ack Request and MRG Block Ack frames. But the tools available like wireshark do not support the interpretation of these frames yet. We decided against developing our own analysis tool chain for this. The results would be corrupted due to the lack of support for the transmission of multiple data frames per TXOP and the rather basic implementation of A-MSDUs in ns-3. In the scenario documented in figure 8.5 the average block size never exceeds 4 or 5 data frames and is on average around 2 data frames. Experiments with additional background traffic showed even worse results. On the one side the background traffic give more information about the channel condition to the SNR-based link adaptation, but on the other side the collisions due to the missing transmission protection in the context of TXOP tends to a degradation in the results.

### 8.2.2 Treatment Of Weak Receives

Our implementation of MRG Block Ack with SNR-based link adaptation works by adapting the physical data rate to the needs of the station with the worst reception conditions. Data frames lost anyhow are retransmitted until their lifetime limit is exceeded. The switch to a more robust physical data rate increases the channel holding time for every transmission. The result of an increase in the number of retransmissions is also an increase of the needed channel holding time. The question is how does a single weak receiver increase the overall channel holding time needed for a group addressed stream?

![Diagram](image)

Figure 8.4: Placing the stations in a line with increasing distance to the access point. Station Wifi_1 can receive with a physical data rate with 54 Mbps. Station Wifi_2 can still receive at a rate with 36 Mbps. Station Wifi_3 needs a physical rate with 18 Mbps. At this low rate the streams are no more feasible with all retransmissions.

The setup of the network is changed by placing the wireless non-AP stations farther away from the access point. As illustrated in figure 8.4 18 Mbps is the resulting
maximal physical data rate in this scenario. But this means that not all trans-
missions and retransmissions needed by non-AP stations to recover from losses can 
be scheduled for transmission. This does not yield an immediate loss of packets. 
Frames not feasible remain in the queue. This causes an increasing delay at the 
application layer. The packet loss occurs as soon as the not feasible frames are 
dropped from the queue. This can happen due to a limited queue size or to the end 
of the packets lifetime. In the simulation nothing happens because queues in ns-3 
have unlimited capacity and the packet lifetime is set to infinity by default. The 
results are illustrated in figure 8.6 on page 114.

If the the weak receiver is ignored to compensate from the throughput under-run at 
the application layer, the situation is changed only to the worse. The treatment of 
the station Wifi_3 with the weak reception by ignoring its need for the data frames 
to be transmitted in a slow but robust physical data rate is illustrated in figure 8.7 
on page 114. The data rate at the application drops significantly for all stations. 
The reason for this observation can be found at the link layer. At the link layer 
the wireless channel is occupied by Stream A and Stream B is deferred. The access 
point requests feedback from the station Wifi_3 for the transmissions under MRG 
Block Ack policy. If the access point does not receive feedback, it repeats its request. 
When the exchange of MRG Block Ack Request and MRG Block Ack frames finally 
succeeds, the remote station Wifi_3 reports packet loss. These lost packets are 
queued for retransmission at the access point, in our simulation. Reducing the 
number of possible retransmissions by changing the packet lifetime does not solve 
the problem. The retransmissions increase the overhead and are lost as the original 
transmission, due to the physical data rate that is selected.

The experiments showed that these retransmissions use up capacity that would 
otherwise be available for possible retransmissions of other packets. As long as the 
packets lifetime is set to a value that allows retransmissions, the weak station uses 
up all this capacity. If the packet lifetime is reduced so that no retransmissions or 
only a small number of retransmissions are possible, the overall throughput at the 
application layer goes down. This happens since the now missing retransmissions are 
also needed by the stations with a better reception need to compensate from their 
packet losses. As long as the access point has to retry packets, it cannot proceed 
with the transmission of the stream. Thus all stations have to wait until the weak 
receiver acknowledges the reception or the packet lifetime expires. Because of these 
observations the simple treatment of a single station by ignoring its needs for a 
more robust data rate does not solve the original problem of throughput under-run 
at the application. Changing the number of possible retransmissions also does not 
solve the problem, because the possibility of retransmissions is also needed by other 
stations.

The first problem is that a weak receiver forces the access point to use a physical 
data rate that does not allow to schedule all transmissions and retransmissions 
needed. The second problem is that a weak receiver causes a huge amount of 
retransmissions which block the progress of the transmission. Tho overcome these 
problems the most simple solution we found is to ignore the non-AP stations with
8.2. TESTS USING MRG BLOCK ACK

a weak reception and also not to ask for feedback. The standard specifies that the access point should ask all stations and repeat data frames until all group members acknowledges their reception or their lifetime expires. Since our implementation does not support the selective not requesting of feedback, this behavior has been simulated by removing the MRG Block Ack agreement for this group member. In our simulation the entries for the non-AP station Wifi_3 have been excluded from the MrgMappingContainer.

Now the access point does not initiate the setup of a MRG Block Ack agreement with the remote station Wifi_3. The access point establish MRG Block Ack agreements with station Wifi_1 for Stream A and B and with station Wifi_2 for Stream B. With station Wifi_3 the access point does not establish any agreement. Now the access point requests only feedback from Wifi_1 and Wifi_2. The packet sinks at the application layer remain unchanged at all three non-AP stations. This setup violates the standard, since MRG Block Ack policy should only be used when the access point has established an MRG Block Ack agreement with every group member. This violation only affects the excluded station as we will see presently.

The measured results in this setup are visualized in figure 8.8 on page 115. For the two stations Wifi_1 and Wifi_2 the measured results at the application layer are in the same magnitude as the results from the first experiment with a shorter distance illustrated in figure 8.5 on page 112. The achieved throughput is on average the transmitted 4Mbit/s. The measured delay and jitter is negligible small. The loss rate is about 1.5%.

For the excluded station Wifi_3 we measure contradicting results. On the one side the achieved throughput exceeds the original transmitted data rate. On the other side the loss rate is with approximately 15% much higher. The reason for this is the missing setup of the data structures for caching and buffering during the setup of the MRG Block Ack agreement at the non-AP station. The station Wifi_3 acts like a legacy station without any support for More Reliable Groupcast could act. The legacy IEEE 802.11 does not specifies feedback or retransmissions of group addressed traffic. A legacy non-AP station is missing a mechanism for the detection of duplicates as the MRG Block Ack defines. This could lead to data frames forwarded up multiple times to the upper layer. The measured throughput overshoot is the result of this. Taking into account the original transmission and its retransmissions. This also means that the measured loss rate is wrong. The actual loss rate is higher. Because of the missing detection of duplicates the IEEE 802.11aa defines that the access point should only transmit data frames using the MRG Block Ack policy if it has a MRG Block Ack agreement with all members of the group. Otherwise the access point should transmit the data frames using the legacy policy without any acknowledgment.

As result from the experiments the conclusion is that as long as it is possible to schedule all transmissions, the access point should use the fastest and least robust physical data rate for group addressed traffic that still enables all group members to receive the transmissions. As soon as the access point is not more possible
to schedule all transmissions using this physical data rate, the access point should switch to a faster and less robust physical data rate and stop addressing the stations for feedback via the MRG Block Ack mechanism that are now no longer able to receive. The determination of the physical data rate that defines if a stations reception is still good enough for being asked for feedback and the number of group addressed streams that are feasible at the same time are important questions for future research.

Figure 8.5: Test of MRG Block Ack with SNR-based link adaptation with distance of 7 meters between each node. Top left to bottom right: throughput, delay, jitter, loss rate at the application layer. Last figure: throughput at the link layer.
8.2. TESTS USING MRG BLOCK ACK

Figure 8.6: Test with 10 meters between each node. The weak receiver Wifi_3 degrade the overall transmission. The not feasible packets remain in the transmission queue, resulting in a low loss rate. Top left to bottom right: throughput, delay, jitter, loss rate at the application layer. Last figure: throughput at the link layer.
Figure 8.7: Test with 10 meters between each node. Ignoring the weak receivers WiFi_3 reception problems grade the overall transmission even more. The higher loss rate cause a higher number of retransmissions at the link layer. Top left to bottom right: throughput, delay, jitter, loss rate at the application layer. Last figure: throughput at the link layer.
8.2. TESTS USING MRG BLOCK ACK

Figure 8.8: Test with 10 meters between each node. The weak receiver Wifi\(_3\) is not part of the MRG Block Ack policy. Thus the access point sets the physical data rate according to the feedback provided by Wifi\(_1\) and Wifi\(_2\). The high throughput measured at Wifi\(_3\) is due to the now missing duplication detection at this station. The increased loss rate shows that Wifi\(_3\) experiences a high loss rate. Top left to bottom right: throughput, delay, jitter, loss rate at the application layer. Last figure: throughput at the link layer.
Chapter 9

Conclusion

In this thesis brief descriptions of the block acknowledgment, admission control and multicast group management in IEEE 802.11 are given. In addition to this admission control schemes as developed at the telecommunications lab are presented. A brief description of the modulation and coding used in IEEE 802.11a was given. The enhanced contention based channel access as defined in IEEE 802.11e is presented together with a brief derivation of the channel holding times for frame transmission.

A summary of link adaptation mechanisms as found in the literature is given. The new More Reliable Groupcast (MRG) policies defined in IEEE 802.11aa are presented with a focus on the MRG Block Ack policy. A SNR-based link adaptation algorithm for group addressed traffic is introduced. The introduced adaptation uses the new MRG Block Ack policy and the measured SNR for the link adaptation and the block size control. An implementation using the ns-3 network simulator was provided. The achieved results using the implementation of the SNR-based link adaptation for group addressed traffic are presented.

9.1 Achieved Enhancement

The experiments with the implementation using the ns-3 network simulator showed that the SNR-based link adaptation using the MRG Block Ack provides a great enhancement compared to the legacy solution. The legacy solution to use the slowest and most robust physical data rate due to the lack of any feedback wasted a lot of channel holding time. The experiments showed that the use of an optimal physical data rate minimizes the channel holding time for group addressed traffic.

The experiment showed that in the legacy situation no more than one audio/visual streaming application with network traffic in the magnitude of an standard IPTV stream is possible at a time using wireless LAN as defined in IEEE 802.11. Using the introduced link adaptation for group addressed traffic multiple IPTV streams are possible at the same time as shown in the experiments.
The applications requirements in terms of throughput, loss rate, delay and jitter are met using the new MRG Block Ack policy as shown in the experiments. The experiments showed that the MRG Block Ack policy can provide a reliable transmission using an automated repeat query that retransmit lost data transmissions. The experiments showed further that as soon as the requested retransmissions are no more feasible using a physical data rate that provides a reliable reception by all group members the weakest group members must be ignored in respect of physical data rate adaptation and requested retransmission. Otherwise it is not possible to fulfill the requirements of the audio/visual application for any group member.

9.2 Simulation Accuracy

The achieved results are in the expected range. The ns-3 network simulator is the designated successor of the ns-2 network simulator and based on it. The accuracy of ns-2 is evaluated in multiple publications. Since the ns-3 is still under development and not all features needed are already implemented or under development. For some open problems a solution was provided in this thesis.

The extensions of the IEEE 802.11 standard defined in IEEE 802.11e and IEEE 802.11n are under development and the current implementation provides only limited support. The EDCA implemented in the ns-3 provides only support for a single frame per TXOP and does not provide a reliable collision avoidance. The aggregation of multiple MSDUs from the upper layer to an aggregated A-MSDU is presented, but lacks support for a possible retransmission using the original separated MSDUs in case of channel errors. Since the MRG Block Ack policy makes implicit use of all these extensions the open problems limit the achieved results. To deal with this and other open problems a revision of the corresponding implementation is planned by the developers of ns-3.

The given implementation of the unicast Block Ack in ns-3 is incomplete. On the receivers side the receiving cache, in order to provide correct feedback to the originator, is still missing. A corresponding cache is provided for the groupcast MRG Block Ack variant. The computation of the channel holding time for the unicast Block Ack control frames was implemented as constant durations optimized for the use with the physical data rates defined in IEEE 802.11a. This is incorrect since the Block Ack control frames can be transmitted using the supported rates unlike other control frames that must be transmitted using the basic physical data rate. A correct implementation taking the variable frame size and the used physical data rate into account is provided.

The current version of ns-3 provides only static routing of multicast traffic in IP networks. No dynamic multicast group membership as defined in the IGMP is provided. Due to the lack of an IGMP implementation, the IGMP snooping at the access point could not be implemented, as proposed in IEEE 802.11aa for the MRG Block Ack. Instead a static group membership mapping was provided. Because of
this no joining or leaving of multicast groups could be simulated, except the initial join after the association of a non-AP station and the access point.

9.3 Further development

The thesis provides a first implementation of the new MRG Block Ack policy. The presented link adaptation is limited due to the network simulation used. From the observations made in the experiments evaluating the implementation and the new extensions defined for the IEEE 802.11 wireless LAN some topics for further development and research arise.

The IEEE 802.11 standard defines that a unique association identifier (AID) is assigned to every station by the access point. The MRG Block Ack use this AID to address the stations that are requested to provide feedback. For this purpose the MRG Block Ack Request frame defines a partial bit field that is organized in octets. In this field every bit represents an AID. The frame contains only a part of the field, starting with the first octet with a bit set up to the last octet with a bit set. Thus the size in octets of the transmitted frames depends on the AIDs assigned to the stations addressed in this frame. The open question is, how could a scheme look like, that use the association management frames to update the distributed AIDs in a way that the resulting MRG Block Ack Request frame is minimal.

The AIDs associated to non-AP stations also defines the order the stations addressed in a MRG Block Ack Request reply. If a different order is wanted, the AIDs need to be reassigned as described above, or multiple MRG Block Ack Requests with different subsets of addressed stations must be sent. This might be needed to implement link adaptation mechanisms, based on MRG Block Ack, that request feedback more often from some stations to compensate different reception conditions, like a hidden station disturbing the reception of a single associated station.

In IEEE 802.11n new physical data rates are defined that use spatial beamforming of the transmitted radio signal to provide high throughput to a single receiver. How could this be used for the transmission of audio/visual applications? Given multiple members of a group addressed traffic stream, could there be cases when it is more efficient to transmit the traffic addressed directly to some stations using the new high throughput modes? Another question is, how could a link adaptation mechanism for unicast and groupcast traffic look like, that take this new high throughput modes into account?

Given such a link adaptation that returns an optimal physical data rate for unicast and group addressed transmission, how could an admission control scheme look like, that takes the problems mentioned above into account? First the optimization of group or direct transmission in the presence of high throughput modes. Second the influence of other parameters, like the AID distribution, on the channel holding time.
The next question is, how this strategies must be changed for scenarios with multiple access points connected with a common wired network in an extended service set (ESS), as defined in the IEEE 802.11 standard.

At last but not least point for further development is the implementation on real hardware. A possibility would be access points using embedded Linux as operation system. Access points implementing the IEEE 802.11n standard and using Linux based operating systems have already reached the market. They provide a basis for the development and evaluation of link admission and admission control schemes, as described, under real conditions.
Bibliography


[23] The ns-3 network simulator. [http://www.nsnam.org/]


